



UNIVERSIDAD COMPLUTENSE MADRID

Diseño y construcción de una impresora 3D

Por Álvaro Velasco García

Dirigido por Juan Jiménez Castellanos

Curso Académico 2018/2019

Trabajo de fin de grado del grado en Ingeniería Informática,
Facultad de Informática, Universidad Complutense de Madrid

Dedicado a Nao.

Agradecimientos

Ha sido un camino muy largo lleno de lecciones, decepciones, alegrías y desesperaciones. En él he conocido a muchas personas y todas ellas han aportado algo al proyecto y sin duda sin ellos no se podría haber llevado a cabo. Sin dejar de lado, claro está, a mi familia que siempre es una constante ayuda. Gracias.

Resumen

Desde los orígenes de la impresión 3D en 1986, fecha en la que se patentó la **estereolitografía** (STL), ha experimentado un crecimiento exponencial [50]. Comenzó siendo una herramienta usada solo en proyecto ingenieriles, arquitectónicos o científicos y ha pasado a poder usarse en el ámbito estudiantil, domestico, etc.

Esto se ha debido principalmente a:

- Vencimiento de algunas patentes
- Desarrollo de la tecnología
- Abaratamiento del sistema de producción de la tecnología.

Con el paso de los años, ésta tecnología acabará integrándose en nuestro día a día y será tan cotidiano como los ordenadores o los teléfonos móviles.

Es un mundo de oportunidades y parece la dirección hacia la que apunta el futuro. Sin embargo la tecnología no ha conseguido difundirse a nivel usuario. Es cierto que la tecnología ha sufrido un descenso drástico en su precio, pero no el suficiente como para que sea asequible para la mayoría de bolsillos. Si nos fijamos en los precios del mercado, vemos como una impresora totalmente funcional, montada y operativa, con una calidad media-baja cuesta alrededor de 900 dólares [1], lo cual si lo juntamos con el hecho de que el usuario medio no ve gran utilidad en el uso de esta tecnología tenemos una combinación muy mala. Es decir, a nivel hogar o personal no parece ser una herramienta muy útil, sigue quedándose en el marco de uso de ingenieros, arquitectos, etc.

Lo que haría falta para que terminara de arrancar esta tecnología y llegara a las cotas de la telefonía o los ordenadores sería buscarle nuevas utilidades o en su defecto desarrollar o encontrar nuevas aplicaciones. Encontrar estas aplicaciones sería más sencillo si existiera una plataforma

completa (diseño estructural, hardware y software) de una impresora fácilmente accesible para los usuarios. De este modo, al ser más sencilla y más accesible para todos, los usuarios podrían colaborar sin tantas dificultades a la hora de encontrar nuevas utilidades. Es por ello que este proyecto tiene por objetivo desarrollar un punto de partida inicial de una impresora 3D unificada (diseño estructural, software y hardware) para poder cumplir los siguientes objetivos:

- Potenciar esta tecnología.
- Sentar las bases de un diseño software-hardware para extender esta tecnología.
- Desarrollar mis aptitudes ingenieriles adquiridas durante la carrera y poner a prueba éstas.

El plan de trabajo es:

- Realizar una etapa de investigación y desarrollo previa para adquirir los conocimientos, tanto teóricos como experimentales que se requieren para llevar a cabo el proyecto.
- Una vez entendidas las diferentes implementaciones posibles tras la etapa de I+D, enfren-
tar de manera experimental los diferentes tipos de implementaciones para dilucidar cuál
de ellas es la óptima.
- Desarrollar el diseño de todo el proyecto y las partes que lo componen, solucionando los
posibles problemas y esquematizando los pasos a seguir para llevar a cabo la implemen-
tación.
- Diseñar un plan económico para evaluar los costes del proyecto.
- Seguir el esquema de implementación desarrollado en la fase de diseño.

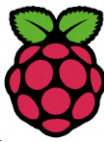
Los entornos de desarrollo que se utilizaron para llevar a cabo el proyecto son los siguientes:

- Apartado estructural de la impresora

- SketchUp  [2]

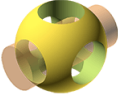
- Apartado hardware

- EasyEDA  [3]

- Raspberry Pi  [4]

- Apartado software

- Eclipse  [5]

- OpenSCAD  [6]

- Ubuntu 16.04  [7]

Palabras Clave

Impresión 3D, Modelado 3D, **RepRap**, Modelado por deposición fundida, Diseño de software para impresión 3D, Diseño de sistema de control para impresoras 3D.

Summary

Since the origins of 3D printing in 1986, the date on which the **estereolitografía** (STL) was patented, it has experienced exponential growth [50]. It began as a tool used only in engineering, architectural or scientific projects and has become available in the student, domestic, etc.

This has been mainly due to:

- Expiration of some patents
- Development of technology
- Abatement of the technology production system.

Over the years, this technology will end up being integrated into our day to day and will be as commonplace as computers or mobile phones.

It is a world of opportunities and it seems the direction towards which the future is pointing. However, technology has not been disseminated at the user level. It is true that technology has suffered a drastic decline in its price, but not enough to make it affordable for most pockets. If we look at the market prices, we see how a fully functional, assembled and operational printer with a medium-low quality costs around 900 dollars [1], which if we put it together with the fact that the average user do not see great utility in the use of this technology we have a very bad combination. That is, at home or personal level does not seem to be a very useful tool, it remains within the framework of the use of engineers, architects, etc.

What it would take to finish to start this technology and reach the heights of telephony or computers would be to seek new utilities or failing to develop or find new applications. Finding these applications would be easier if there was a complete platform (structural design, hardware and software) of a printer easily accessible to users. In this way, as it is easier and more accessible for all, users could collaborate without so much difficulty when it comes to finding

new utilities. That is why this project aims to develop an initial starting point for a unified 3D printer (structural design, software and hardware) to meet the following objectives:

- Enhance this technology.
- Build a foundations of a software-hardware design to extend this technology.
- Develop my engineering skills acquired during the race and put these to the test.

The work plan is:

- Perform a stage of research and prior development to acquire the knowledge, both theoretical and experimental, that are required to carry out the project.
- Once the different possible implementations have been understood after the R and D stage, face experimentally the different types of implementations to elucidate which of them is the optimal one.
- Develop the design of the whole project and the parts that compose it, solving the possible problems and outlining the steps to follow to carry out the implementation.
- Design an economic plan to evaluate the costs required to carry out the project.
- Follow the implementation scheme developed in the design phase.

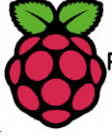
The development environments that were used to carry out the project are the following:

- Structural section of the printer

- SketchUp  [2]

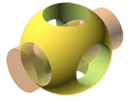
- Hardware section

- EasyEDA  [3]

- Raspberry Pi  RaspberryPi [4]

■ Software section

- Eclipse  eclipse [5]

- OpenSCAD  OpenSCAD [6]

- Ubuntu 16.04  [7]

Keywords

3D printing, 3D modeling, **RepRap**, Molded deposition modeling, Software design for 3D printing, Control system design for 3D printers.

Índice general

Lista de figuras	19
Lista de tablas	23
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Estructura del trabajo	2
1.4. Motivation	5
1.5. Goals	6
1.6. Structure of memory	6
2. Estado del arte	9
2.1. Definición	9
2.2. Un poco de historia	9
2.3. Técnicas	10
2.3.1. Estereolitografía:STL	11
2.3.2. Sintetizado selectivo por láser:SLS	12
2.3.3. Fabricación mediante balística de partículas:BPM	14
2.3.4. Fabricación laminada:LOM	15
2.3.5. Impresión 3D:3DP	15
2.3.6. Modelado por deposición fundida:FDM	16
2.4. Estructura general de una impresora FDM	17
2.4.1. Descripción general	17
2.4.2. Marco estructural	18

2.4.3.	Plataforma	21
2.4.4.	Extrusor o HotEnd	21
2.4.5.	Controladores	22
2.4.5.1.	Motores de continua	23
2.4.5.2.	Motores bipolares	24
2.4.6.	Placa	24
2.5.	Proceso de Impresión	26
2.5.1.	Diseño 3D	26
2.5.1.1.	Modelado 3D	27
2.5.1.2.	Escaneo tridimensional	27
2.5.2.	Traducción del modelo	28
2.5.3.	Impresión	30
2.6.	La impresión 3D ahora	30
2.6.1.	Alimentación y farmacia	31
2.6.2.	Ámbito domestico	31
2.6.3.	Aeronáutica	32
2.6.4.	Automoción	32
2.6.5.	Deportes	33
2.6.6.	Medicina y biología	33
3.	Fase uno: Diseño estructural	35
3.1.	I+D previo	35
3.1.1.	Primera versión	36
3.1.2.	Segunda versión	37
3.1.3.	Tercera versión	39
3.2.	Prototipo definitivo	41
3.2.1.	Aplicación del conocimiento adquirido en la fase de I+D	42
3.2.2.	Partes de la impresora y lista de materiales	44
3.2.2.1.	Vistas acotadas del marco estructural	44
3.2.2.2.	Vistas acotadas del eje X	46
3.2.2.3.	Vistas acotadas del eje Y	48

<i>ÍNDICE GENERAL</i>	15
3.2.2.4. Vistas acotadas del eje Z	49
3.2.2.5. Materiales	52
4. Fase dos: Diseño hardware	53
4.1. Componentes	53
4.2. Implementación	55
4.2.1. RaspberryPi	55
4.2.2. Nema 17	56
4.2.3. DRV8825	59
4.2.3.1. Alimentación y avance rotacional por paso	63
4.2.3.2. Control del motor	63
4.2.4. MCP3002	64
4.2.5. CMX100D10	67
4.2.6. Termistor NTC 100K	69
4.2.7. Ventiladores	71
4.2.8. Botones	72
4.2.9. Detector de filamento	73
5. Fase tres: Diseño software	77
5.1. Slicer	77
5.1.1. ¿Dónde opera?	79
5.1.2. Estructura y composición	80
5.1.2.1. BaseUniTest	80
5.1.2.1.1. Clase Test	81
5.1.2.1.2. Clase BaseUniTest	81
5.1.2.2. Configuration	83
5.1.2.3. Figure	84
5.1.2.4. Geometry	85
5.1.2.4.1. Equations	85
5.1.2.4.2. Line	85
5.1.2.4.3. Plane	88
5.1.2.4.4. Point	88

5.1.2.4.5. Triangle	90
5.1.2.4.6. Vector	94
5.1.2.5. Main	95
5.1.2.6. Math	95
5.2. Nao	96
5.2.1. ¿Dónde opera?	96
5.2.2. Estructura y composición	97
5.2.2.1. Common	97
5.2.2.2. Controllers	97
5.2.2.3. Main	98
5.2.2.4. Models	98
5.2.2.5. Views	99
5.2.2.6. Carpetas	100
5.2.3. Mapa de acciones	101
5.2.3.1. Cargar una figura	102
5.2.3.2. Añadir y eliminar una impresora	103
5.2.3.3. Imprimir	104
5.2.4. Comunicación y procesos paralelos	105
5.2.4.1. CheckStatus	105
5.2.4.2. CheckInterruptions	106
5.2.4.3. AddPrinter	107
5.3. Line printer	108
5.3.1. ¿Dónde opera?	108
5.3.2. Estructura y composición	109
5.3.2.1. Clase Agent	110
5.3.2.2. Paquete Interrupt	110
5.3.2.3. Paquete Thermistor	110
5.3.2.4. Paquete Stepper	112
5.3.2.5. Paquete Boss	112
5.3.3. Carpetas	113

<i>ÍNDICE GENERAL</i>	17
6. Apartado económico	115
7. Aplicación de conocimientos	119
8. Conclusiones	121
Glosario	129
Bibliografía	131

Índice de figuras

2.1. Gráfica de ventas de impresoras en los últimos años. [8]	10
2.2. Procedimiento de impresión 3D por estereolitografía (STL). [50]	11
2.3. Voladizo en una figura.	13
2.4. Procedimiento de impresión 3D por sinterización selectiva por láser (SLS). [50]	14
2.5. Procedimiento de impresión 3D por deposición fundida (FDM).	17
2.6. Ejes de una impresora 3D FDM : Modelo PRUSA.	18
2.7. Extrusor o HotEnd: modelo mk8.	22
2.8. Esquema de funcionamiento de un motor de continua.	23
2.9. Partes de una impresora 3D FDM : Modelo PRUSA.	26
2.10. Ejemplo de triangulación de nube de puntos de una esfera en OpenSCAD. . . .	28
2.11. Ejemplo de archivo de texto que contiene la información de una figura en formato .STL: OpenSCAD.	29
2.12. Rebanamiento de figura en sentido del eje Z.	30
3.1. Primera versión de la impresora. Foto tomada durante la prueba de funcionalidad.	37
3.2. Segunda versión de la impresora. Foto tomada durante la prueba de rendimiento y funcionalidad.	38
3.3. Tercera versión de la impresora.	39
3.4. Vista de planta de la impresora.	41
3.5. Vista alzada de la impresora.	42
3.6. Vista de perfil la impresora.	42
3.7. Vista acotada desde arriba del marco estructural.	44
3.8. Vista desde abajo del marco estructural.	44
3.9. Vista acotada frontal del marco estructural.	45

3.10. Primera vista acotada desde arriba del eje X.	46
3.11. Segunda vista acotada desde arriba del eje X.	46
3.12. Tercera vista acotada desde arriba del eje X.	47
3.13. Cuarta vista acotada desde arriba del eje X.	47
3.14. Vista acotada frontal del eje X.	48
3.15. Vista desde arriba del eje Y	48
3.16. Vista frontal del eje Y	49
3.17. Vista acotada desde arriba del eje Z.	49
3.18. Vista acotada desde atrás del eje Z.	50
3.19. Vista acotada izquierda del eje Z.	51
3.20. Vista acotada derecha del eje Z.	51
4.1. Esquema completo de la placa de control.	55
4.2. BCM Raspberry en la placa.	56
4.3. Motor bipolar Nema 17.	57
4.4. Esquema simplificado de las bobinas de un motor bipolar.	58
4.5. Motor nema 17 con sus bobinas y rotor.	58
4.6. Simplificación de un puente en H.	59
4.7. Simplificación de un doble puente en H.	60
4.8. Esquema del DRV8825.	62
4.9. Ubicación de los variadores en la placa.	62
4.10. Esquema del MCP3002.	64
4.11. MCP3002 en la placa de control de la impresora.	65
4.12. Esquema simplificado de la comunicación entre el esclavo MCP3002 y la controladora.	67
4.13. Esquema simplificado de un relé. El circuito recibe corriente.	68
4.14. Relé en la placa de control de la impresora.	68
4.15. Diodo en la placa de control de la impresora.	69
4.16. Conexiones del termistor en la placa de control de la impresora.	70
4.17. Divisor de tensión en la placa de control de la impresora.	71
4.18. Ventiladores en la placa de control de la impresora.	72

4.19. Pulsadores en la placa de control de la impresora.	73
4.20. Detector de filamento activo.	74
4.21. Detector de filamento desactivado.	74
4.22. Conexiones del detector de filamento en la placa de control de la impresora. . .	75
5.1. Planos de rebanamiento.	79
5.2. Cálculo de los planos que cortan al triángulo($APRZ = 1$).	79
5.3. Lugar que ocupa Slicer dentro del esquema general de sistema.	80
5.4. Ejemplo de Test.	82
5.5. Ejemplo de ejecución satisfactoria.	82
5.6. Ejemplo de ejecución insatisfactoria.	83
5.7. Ejemplo de ordenación de rectas.	87
5.8. Ejemplo de puntos que componen una recta.	87
5.9. Ejemplo de comparación entre rectas.	88
5.10. Casos de posición relativa entre plano y triángulo.	90
5.11. Paso 1: Intersección normal entre plano de slice y triángulo.	91
5.12. Paso 2: Intersección normal entre plano de slice y triángulo.	92
5.13. Paso 3: Intersección normal entre plano de slice y triángulo.	92
5.14. Paso 1: Intersección especial entre plano de slice y triángulo.	93
5.15. Paso 2: Intersección especial entre plano de slice y triángulo.	93
5.16. Paso 3: Intersección especial entre plano de slice y triángulo.	94
5.17. Ejemplo del flujo que sigue el patrón modelo-vista-controlador.	100
5.18. Proceso de carga de una figura.	102
5.19. Proceso para añadir una impresora.	103
5.20. Proceso para imprimir una figura.	104
5.21. Esquema de funcionamiento normal del proceso CheckStatus.	106
5.22. Esquema de funcionamiento normal del proceso CheckInterruptions.	107
5.23. Lugar que ocupa Line printer en el esquema general.	108
5.24. Simplificación del circuito de lectura de temperaturas.	111
5.25. Formato del archivo characteristics.	114
5.26. Formato del archivo configuration.	114

5.27. Formato del archivo registro de actividad del servidor.	114
---	-----

Índice de cuadros

3.1. Lista de materiales de la estructura.	52
4.1. Lista de componentes que comprenden la placa.	54
4.2. Estados de un motor paso a paso.	61
6.1. Lista de materiales de la estructura y sus costes.	116
6.2. Lista de componentes que comprenden la placa y sus costes.	117
6.3. Coste total del proyecto.	118

Capítulo 1

Introducción

1.1. Motivación

Desde los orígenes de la impresión 3D en 1986, fecha en la que se patentó la **estereolitografía** (STL), ha experimentado un crecimiento exponencial [50]. Comenzó siendo una herramienta usada solo en proyecto ingenieriles, arquitectónicos o científicos y ha pasado a poder usarse en el ámbito estudiantil, domestico, etc.

Esto se ha debido principalmente a:

- Vencimiento de algunas patentes
- Desarrollo de la tecnología
- Abaratamiento del sistema de producción de la tecnología.

Con el paso de los años, ésta tecnología acabará integrándose en nuestro día a día y será tan cotidiano como los ordenadores o los teléfonos móviles.

Es un mundo de oportunidades y parece la dirección hacia la que apunta el futuro. Sin embargo la tecnología no ha conseguido difundirse a nivel usuario. Es cierto que la tecnología ha sufrido un descenso drástico en su precio, pero no el suficiente como para que sea asequible para la mayoría de bolsillos. Si nos fijamos en los precios del mercado, vemos como una impresora totalmente funcional, montada y operativa, con una calidad media-baja cuesta alrededor de 900 dólares [1], lo cual si lo juntamos con el hecho de que el usuario medio no ve gran utilidad en el uso de esta tecnología tenemos una combinación muy mala. Es decir, a nivel hogar o personal

no parece ser una herramienta muy útil, sigue quedándose en el marco de uso de ingenieros, arquitectos, etc.

Lo que haría falta para que terminara de arrancar esta tecnología y llegara a las cotas de la telefonía o los ordenadores sería buscarle nuevas utilidades o en su defecto desarrollar o encontrar nuevas aplicaciones. Encontrar estas aplicaciones sería más sencillo si existiera una plataforma completa (diseño estructural, hardware y software) de una impresora fácilmente accesible para los usuarios. De este modo, al ser más sencilla y más accesible para todos, los usuarios podrían colaborar sin tantas dificultades a la hora de encontrar nuevas utilidades. Es por ello que este proyecto tiene por objetivo desarrollar un punto de partida inicial de una impresora 3D unificada (diseño estructural, software y hardware) para poder cumplir una serie de objetivos.

1.2. Objetivos

El proyecto tiene como objetivo principal el diseño de un prototipo de impresora 3D desde cero y tiene por objetivos secundarios:

- Sentar las bases de un diseño software-hardware para extender esta tecnología.
- Potenciar esta tecnología
- Desarrollar mis aptitudes ingenieriles adquiridas durante la carrera y poner a prueba éstas.

1.3. Estructura del trabajo

La memoria estará dividida en nueve capítulos.

- **Capítulo uno:** Introducción del proyecto.
- **Capítulo dos:** Consistirá en una exposición del estado del arte de la impresión 3D. Como ha ido evolucionando y su estado actual hoy en día. También incluye un análisis general de la composición de una impresora 3D, las técnicas usadas, sus ventajas y desventajas, el proceso de impresión y las futuras aplicaciones que podrían tener las impresoras 3D.

- **Capítulo tres:** Consiste en una explicación de una de las tres partes que componen éste proyecto, el **diseño**. En el se contará como ha ido evolucionando el diseño de la impresora, desde su estadio más primitivo hasta la versión actual.
- **Capítulo cuatro:** Es la segunda parte que compone el diseño de la impresora, el **hardware**. En ella se especificará las características del circuito diseñado, los componentes usados y la relación entre ellos.
- **Capítulo cinco:** Será el último capítulo que compone el diseño, el **software**. En éste se explicará detalladamente la implementación de código y que papel juega cada componente de éste.
- **Capítulo seis:** Contendrá un resumen económico del desarrollo del proyecto.
- **Capítulo siete:** Se mostrara en qué partes del proyecto se han aplicado los conocimientos adquiridos durante el desarrollo de los estudios.
- **Capítulo ocho:** Contendrá las conclusiones a las que se ha llegado una vez finalizó todo el desarrollo del proyecto así como una crítica a éste, sus ventajas e inconvenientes. Además contendrá las posibles mejoras que se podrían realizar.

Introduction

1.4. Motivation

Since the origins of 3D printing in 1986, the date on which the **estereolitografía**(STL) was patented, it has experienced exponential growth [50]. It began as a tool used only in engineering, architectural or scientific projects and has become available in the student, domestic, etc.

This has been mainly due to:

- Expiration of some patents
- Development of technology
- Abatement of the technology production system.

Over the years, this technology will end up being integrated into our day to day and will be as commonplace as computers or mobile phones.

It is a world of opportunities and it seems the direction towards which the future is pointing. However, technology has not been disseminated at the user level. It is true that technology has suffered a drastic decline in its price, but not enough to make it affordable for most pockets. If we look at the market prices, we see how a fully functional, assembled and operational printer with a medium-low quality costs around 900 dollars [1], which if we put it together with the fact that the average user do not see great utility in the use of this technology we have a very bad combination. That is, at home or personal level does not seem to be a very useful tool, it remains within the framework of the use of engineers, architects, etc.

What it would take to finish to start this technology and reach the heights of telephony or computers would be to seek new utilities or failing to develop or find new applications. Finding these applications would be easier if there was a complete platform (structural design,

hardware and software) of a printer easily accessible to users. In this way, as it is easier and more accessible for all, users could collaborate without so much difficulty when it comes to finding new utilities. That is why this project aims to develop an initial starting point for a unified 3D printer (structural design, software and hardware) to be able to fulfill a series of objectives.

1.5. Goals

The main objective of the project is to project the prototype of a 3D printer from scratch and its secondary objectives are:

- Enhance this technology.
- Build a foundations of a software-hardware design to extend this technology.
- Develop my engineering skills acquired during the race and put these to the test.

1.6. Structure of memory

The memory will be divided into nine chapters.

- **Chapter one:** Introduction of the project.
- **Chapter two:** Consists of an exhibition of the state of the art of 3D printing. As it has been evolving and its current state today. It also includes a general analysis of the composition of a 3D printer, the techniques used, their advantages and disadvantages, the printing process and the future applications that 3D printers could have.
- **Chapter three:** It consists of an explanation of one of the three parts that make up this project, the textbf design. In it will be told how the design of the printer has evolved, from its most primitive stage to the current version.
- **Chapter four:** It is the second part that makes up the design of the printer, textbf hardware. It will specify the characteristics of the circuit designed, the components used and the relationship between them.

- **Chapter five:** It will be the last chapter that composes the design, textbf software. This will explain in detail the implementation of code and what role each component plays.
- **Chapter six:** It will contain an economic summary of the development of the project.
- **Chapter seven:** It will be shown in which parts of the project the knowledge acquired during the development of the studies has been applied.
- **Chapter eight:** It will contain the conclusions that have been reached once the entire development of the project has ended as well as a critique of this, its advantages and disadvantages. It will also contain the possible improvements that could be made.

Capítulo 2

Estado del arte

Este capítulo está basado en los primeros capítulos del libro Impresión 3D de Sergio Gómez González [50]:

2.1. Definición

La impresión 3D es un conjunto de tecnologías de fabricación aditiva en las que el objeto a construir se crea por descomposición de un material capa a capa a partir de un **modelo 3D**. La forma en la que se fabrica el modelo define el tipo de tecnología de impresión 3D y se las denomina técnicas de impresión. Algunas de estas técnicas consisten en calentar el material antes de su deposición(**FDM**), en la fundición del material(**SLS**), que en general suele ser metálico o se deposita en forma de láminas delgadas y cortadas(**LOM**). En la sección 3.3 se explicarán las técnicas con mayor profundidad y variedad.

2.2. Un poco de historia

Los inicios de la impresión 3D se remontan a principios de los años 80, cuando Chuck Hull fundó la compañía *3D Systems* y patentó la **estereolitografía**. A partir de entonces la evolución durante los últimos 40 años ha sido imparable. La patente de Hull describe cómo un haz de luz ultravioleta se focalizaba sobre la superficie de una cuba rellena de líquido **fotopolímero**. Los rayos de luz dibujan el objeto en la superficie del líquido, capa a capa, usando fotopolimerización (o cross-linking) para crear el sólido. En 1986, Hull fundó la primera compañía que

generalizaba y comercializaba esta técnica.

El movimiento Maker (comunidad **RepRap**) y la filosofía **DIY** han favorecido que las universidades, las pequeñas empresas y particulares se fabriquen sus propias impresoras a precios razonables.

En 2012 se vendieron tantas impresoras en el mundo como en el período comprendido entre 1980 y 2011. En la figura 2.1 se puede ver que esta tendencia de ventas no ha parado de aumentar año tras año. Actualmente, las impresoras personales se pueden adquirir a precios inferiores a 1000 euros y en prácticamente cualquier tienda de informática.

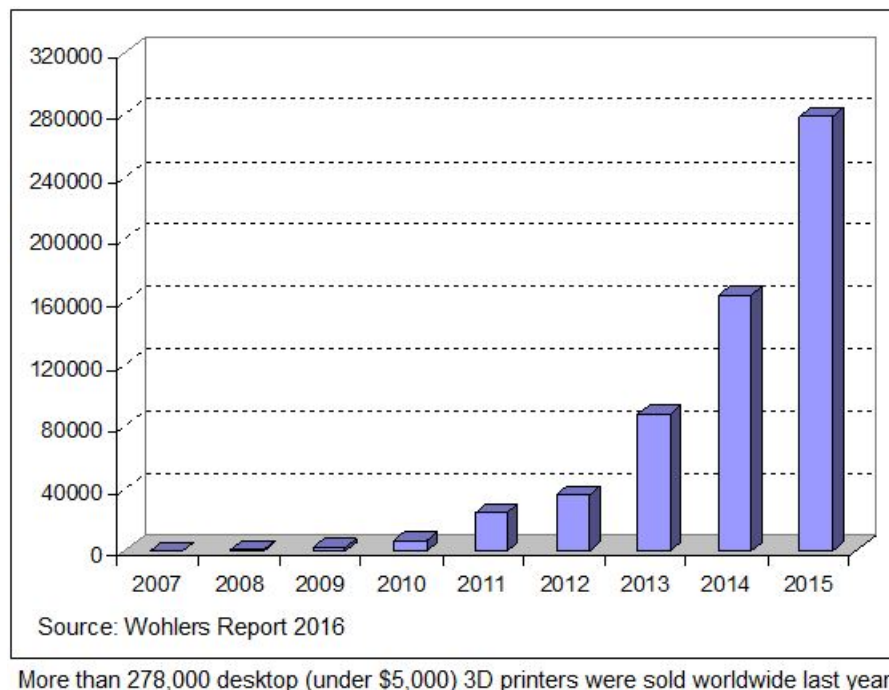


Figura 2.1: Gráfica de ventas de impresoras en los últimos años. [8]

2.3. Técnicas

La forma en la que se fabrica el modelo define el tipo de tecnología de impresión 3D y se las denomina técnicas de impresión. A continuación se expondrán las diferentes técnicas más extendidas existentes en la actualidad.

2.3.1. Estereolitografía:STL

La pieza que se imprime se construye capa a capa a partir de la solidificación de una resina líquida por la incidencia de un haz de luz. El láser sigue las secciones capa a capa definidas en el **modelo 3D** y consigue solidificarlo. El material suele ser un polímero y éste en general está contenido en una cubeta sobre la que se desplaza una plataforma con un movimiento descendente. El movimiento de la plataforma coincide con el espesor de las capas definidas según la resolución a obtener. Cuando termina el proceso de impresión el prototipo en construcción requiere de una fase de horneado. En la figura 2.2 se muestra un esquema gráfico de una impresora que imprime con esta técnica.

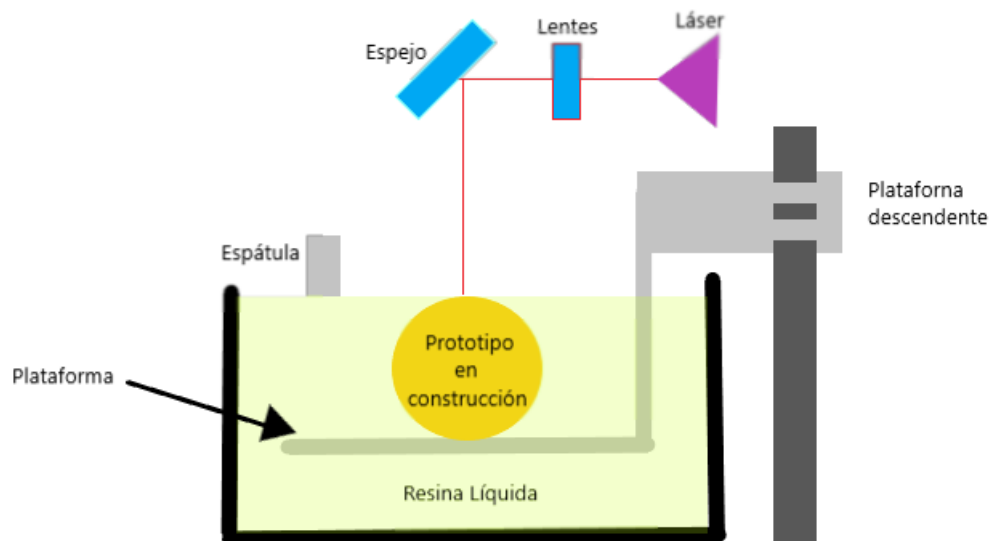


Figura 2.2: Procedimiento de impresión 3D por **estereolitografía** (STL). [50]

Ventajas:

- Consigue piezas con un buen acabado superficial.
- Se obtiene una buena precisión dimensional.
- Admite geometrías complejas.
- Admite espesores de pared finos.
- Permite un gran nivel de definición de detalles.

Inconvenientes:

- La precisión depende de la calidad de la resina.
- La precisión depende de la definición del espesor de la capa que éste a su vez depende de la calidad de la resina usada.
- La pieza corre riesgo de sufrir deformaciones en su última fase cuando es horneada.
- La creación de soportes con voladizos pueden dejar marcas en la superficie de la pieza.
- Los prototipos son frágiles y translúcidos.

2.3.2. Sintetizado selectivo por láser:SLS

El procedimiento de construcción se realiza capa a capa a partir de la fusión, por el aporte energético del láser, de los granos del material en forma de polvo fino depositado y comprimido por un rodillo. El láser recorre las secciones de cada una de las capas a solidificar y calienta el polvo por encima de su temperatura de **sinterización** provocando la unión de los granos. Las zonas o regiones por las que el láser no pasa deja el polvo suelto, que se puede desprender fácilmente y se utiliza como elemento de soporte para la **sinterización** de las capas sucesivas, por lo que la técnica no precisa de estructuras de soporte en la construcción de modelos con voladizos(2.3). En la figura 2.4 se muestra el esquema general de una impresora que sigue esta técnica.



Figura 2.3: **Voladizo** en una figura.

Ventajas:

- Se puede imprimir en una gran cantidad de materiales: polímeros, metales y cerámicas.
- Gran precisión y reproducción de geometrías complejas sin necesidad de soportes ni eliminación de materiales sobrantes.
- Los modelos resultantes suelen tener buenas **propiedades mecánicas**.

Inconvenientes:

- Las impresoras que aplican esta técnica tienen unos precios bastante elevados.
- Esta técnica es bastante lenta y los modelos tardan mucho en realizarse.
- Las piezas tienen una superficie porosa, lo cual puede ser un inconveniente en ciertas piezas que requieran poca fricción.

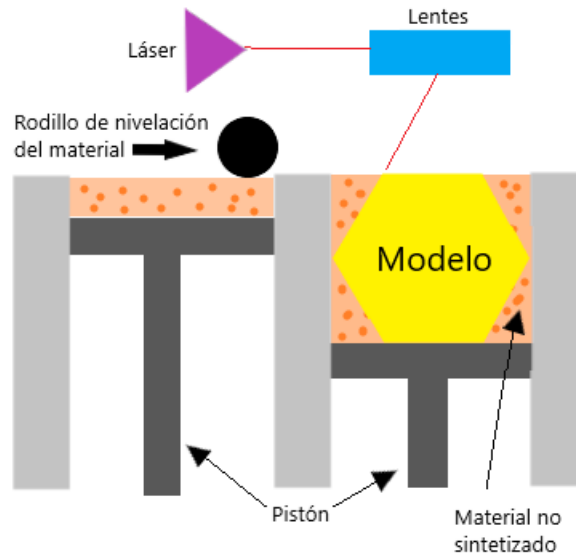


Figura 2.4: Procedimiento de impresión 3D por **sinterización** selectiva por láser (**SLS**). [50]

2.3.3. Fabricación mediante balística de partículas: **BPM**

Con la técnica **BPM**, el material en estado fundido se proyecta en forma de gotas muy pequeñas (pulverizadas) siguiendo la sección de cada una de las capas. El impacto de las gotas sobre la superficie consigue la solidificación. El proceso se realiza en vacío o en atmósfera de nitrógeno (N₂) para evitar la oxidación.

Ventajas:

- Es económico y ecológico.
- No requiere de soportes ni de fase de horneado.

Inconvenientes:

- Las piezas obtenidas son bastante frágiles.
- La necesidad de una atmósfera inerte hace difícil su uso en lugares convencionales, esto dificulta su comercialización.
- Se dispone de poca variedad de materiales para la fabricación de piezas.

2.3.4. Fabricación laminada:LOM

En la técnica **LOM**, la obtención del modelo se realiza capa a capa a partir de la adhesión de láminas, unas sobre otras, y el recorte de las mismas mediante haz de láser (CO2) o cuchillas siguiendo los contornos definidos por el **modelo 3D**. Se utiliza papel autoadhesivo plástico alimentado por rollo que se pega en la capa anterior. En general suele dejar papel sobrante.

Ventajas:

- Es bastante rápido.
- Tiene gran variedad de materiales disponibles.
- No requiere fase de horneado.
- No son necesarios soportes para la estructura de la impresora.

Inconvenientes:

- Las piezas impresas no tienen un buen acabado superficial.
- Es difícil definir huecos interiores.
- Se desperdicia gran cantidad de material.

2.3.5. Impresión 3D:3DP

Las impresoras que usan la técnica **3DP** obtienen el modelo a partir de la adición, capa a capa, de un material en forma de resina o polvo (polímero, cerámica o metal) que se compacta y cohesiona por la deposición de un **aglutinante** siguiendo distintas secciones del **modelo 3D**.

Ventajas:

- Gran velocidad.
- Posibilidad de imprimir en varios colores.

Inconvenientes:

- Poca precisión.
- Baja calidad.

- Necesidad de fase de horneado.
- En algunos casos requiere de soportes de impresión.

2.3.6. Modelado por deposición fundida:FDM

Consiste en depositar un material fundido (ABS, PLA o PA) previamente por el extrusor sobre otras capas ya frías depositadas anteriormente con el fin de garantizar su rápida solidificación y endurecimiento. La deposición del material fundido se realiza a través del extrusor que recibe filamento de la bobina. En la figura 2.5 se muestra un esquema general de las impresoras que siguen esta técnica.

Ventajas:

- Es una de las técnicas de bajo coste más usadas y extendidas.
- Hay una gran variedad de materiales disponibles.
- Tiene unos precios muy competitivos en relación con otras impresoras que usan otras técnicas.
- No produce residuos.
- La impresión de piezas huecas o muy porosas es muy rápida.

Inconvenientes:

- Necesita de soportes o estructuras de apoyo.
- Necesita caras de apoyo para las partes inclinadas o voladizas de la pieza extruida.
- Es bastante lenta si se desean **extruir** piezas macizas y de gran tamaño.

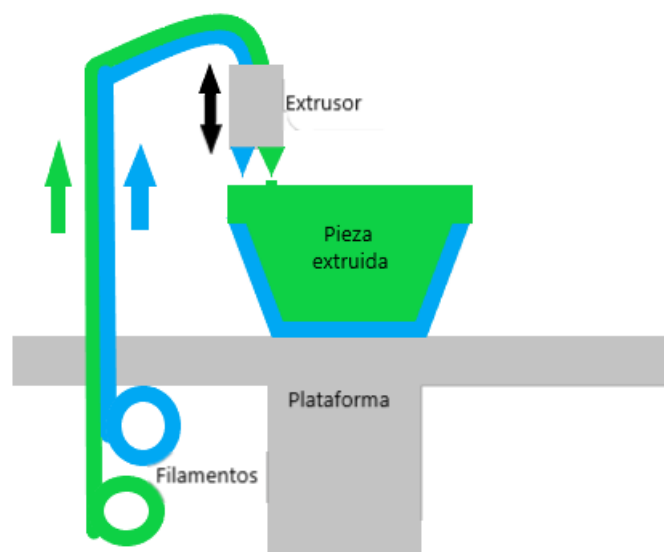


Figura 2.5: Procedimiento de impresión 3D por deposición fundida (FDM).

Aparte de todas las técnicas mencionadas anteriormente, existen algunas otras:

- Solid Ground Curing (SGC) [9].
- Electron Beam Melting (EBM) [10].
- Shape Melting o inyección sólida por impresión (SPI) [11].
- Thermal polymerization o polimerización térmica de líquidos (TP) [12].

2.4. Estructura general de una impresora FDM

La impresora que se implementará en éste proyecto estará basada en la técnica **FDM** mencionada anteriormente debido a su buen balance entre ventajas e inconvenientes, su gran extensión y a ser la más viable económicamente con mis posibilidades.

2.4.1. Descripción general

Estas impresoras están formadas por motores paso a paso para mover los ejes X, Y, Z, motores para mover el extrusor, correas, poleas, rodamientos, varillas roscadas y lisas, tuercas, tornillos, endtops o topes para los ejes, plancha de vidrio para la base y HotEnd o extrusor para

la fusión del plástico. En las figuras 2.6 y 2.9 se muestran de manera general las partes que componen una impresora.

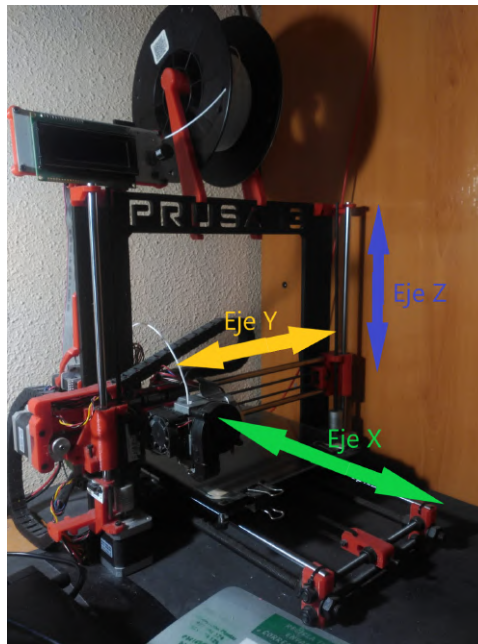


Figura 2.6: Ejes de una impresora 3D **FDM** : Modelo PRUSA.

2.4.2. Marco estructural

El marco estructural es la estructura externa de la impresora 3D y da estabilidad mecánica. Se fabrica en diferentes materiales, madera, aluminio, etc. La mayoría de fabricantes o usuarios ofrecen piezas que forman la estructura en un mismo **modelo 3D** para poder fabricar recambios en caso de requerirlos en el futuro.

En general, es recomendable que los materiales que componen el marco estructural sean ignífugos y tengan buena tolerancia al calor ya que los extrusores pueden alcanzar temperaturas cercanas a los 300°, dependiendo del polímero del que esté hecho el filamento.

Estos son los polímeros más convencionales usados en la impresión de figuras [13]:

- **ABS** Acrilonitrilo Butadieno Estireno, es el más barato entre los plásticos más utilizados. Tiene una gran versatilidad permitiendo su tratamiento post impresión y sus **propiedades mecánicas** le confieren una gran resistencia y durabilidad en el tiempo. Tiene una temperatura de fusión desde los 215° hasta los 250° y requiere de un aporte extra de calor durante la extrusión para asegurar la unión entre las diferentes capas, éste calor se aporta

a través de la cama caliente que en el caso del ABS requiere que este entre los 90° hasta los 103°. Este material no es biodegradable pero sí es reciclable. Si se usa este material, la impresora debe estar en un lugar abierto ya que emana gases no muy nocivos pero sí perjudiciales.

- **PLA** Ácido Poliláctico o Poliláctido. Este polímero proviene de un derivado del almidón el cual se obtiene de materia orgánica como es el maíz o las patatas, de modo que este plástico es biodegradable, tardando aproximadamente 100 años en biodegradarse. También es reciclable y no requiere de aporte calórico extra para su uso. Su rango de temperaturas de fusión está entre los 160° y los 230°. La parte negativa de este material es que mecánicamente hablando es poco resistente y tendente a la fractura, no aconsejable para fabricar piezas que deban soportar una gran tensión mecánica.
- **PVA** Alcohol Polivinilo, es un polímero que tiene la característica de ser soluble en agua, teniendo una gran capacidad de absorción de ésta. Este tipo de polímero suele usarse para construir estructuras de apoyo para las partes voladizas de las figuras, ya que, una vez finalizada la impresión de la figura, resulta sencillo eliminar estas estructuras de apoyo con agua. Este material es biodegradable y tiene una temperatura de fusión entre los 180° y los 200°.
- **PC** Policarbonato. Este es el material, de entre los polímeros más convencionales, que mayor temperatura de fusión tiene, entre los 280° y los 305°. También requiere de una cama caliente que debe estar entre los 85° y los 95°. Es reciclable pero no biodegradable.
- **Soft PLA** Ácido Poliláctico flexible, es un derivado del PLA el cual tiene la cualidad mecánica de ser flexible además de ser biodegradable y reciclable. Requiere de una temperatura de fusión de entre 200° y los 220° y de una cama caliente que debe tener una temperatura cercana a los 60°.
- **HDPE** Polietileno de alta densidad, este tiene la cualidad de ser insoluble en casi todos los disolventes y pegamentos, es reciclable pero no biodegradable, no requiere de cama caliente y su temperatura de fusión se encuentra entre los 225° y los 230°. Además este polímero tiende a encogerse lo cual da ciertos problemas de deformación de las figuras resultantes.

- **Nailon** Da un buen acabado a las piezas fabricadas con él, minimizando las superficies porosas de éstas, es reciclable pero no biodegradable. El Nailon tiende a encogerse pero algunos fabricantes de filamento han conseguido reducir este efecto. No requiere de cama caliente y su temperatura de fusión es de entre 240° y 250°.
- **Laybrick** Material compuesto por una mezcla de polímero y polvo de yeso, lo cual le permite tener una textura diferente en función de la temperatura a la que se extruya. Su temperatura de extrusión está entre los 160° y los 220° y no requiere de cama caliente. Cuanto mayor sea la temperatura a la que se extruye el material mayor rugosidad tendrá la superficie de la figura, cuanto menor sea la temperatura más lisa será ésta.
- **Laywood-D3** Este material es una mezcla de polímero y polvo de madera. Tiene unas temperaturas de fusión entre 175° y 250°, no requiere de cama caliente y el filamento, antes de **extruirse**, es bastante frágil. Si se extruye a mayor temperatura más se oscurece el material.
- **Termocromáticos y fotocromáticos** Estos materiales tienen la propiedad de cambiar de color en función de la temperatura que tengan, termocromáticos, y en función de la luz que reciban, fotocromáticos. Tienen una temperatura de fusión entre 210° y 230°, no requieren de cama caliente, son biodegradables y reciclables.
- **Fluorescentes** Estos polímeros tienen la propiedad de brillar en la oscuridad, son biodegradables y reciclables. Requieren de una temperatura de cama caliente cercana a los 60° y tienen una temperatura de fusión de entre los 160° y 230°.
- **T-Glass** Es un material que emula el cristal, siendo un material transparente. No es biodegradable pero sí reciclable. Es recomendable usar una cama caliente para que el material no se enfríe de golpe y tiene unas temperaturas de fusión comprendidas entre los 212° y los 224°.
- **Filaflex** Este tipo de polímero tiene la **propiedad mecánica** de ser muy elástico. Requiere de una temperatura de fusión de entre los 220° y los 230° y no requiere de cama caliente. También cabe destacar que es resistente a ciertos disolventes.

2.4.3. Plataforma

Es la superficie plana sobre la cual se imprimen las piezas, capa a capa. Al igual que el marco estructural, es recomendable que este sea de un material ignífugo y tengan buena tolerancia al calor por los mismos motivos. Las más avanzadas están calefactadas o soportan un elemento calefactado (llamado comúnmente cama calefactada) para mejorar la adhesión o pegado de las diversas capas.

Esta plataforma está guiada por varillas roscadas, poleas y por motores para su desplazamiento a lo largo del eje X. En algunas impresoras la plataforma tiene un movimiento descendente en el eje Z y en otras la plataforma siempre permanece fija a la misma altura siendo el extrusor el que se desplaza en el eje Z durante el proceso de impresión.

2.4.4. Extrusor o HotEnd

Se encarga de calentar el plástico/polímero fundido a partir de un filamento continuo. En general suele calentarse gracias a una resistencia calefactora que se encuentra en el interior del extrusor aunque también hay algunos que utilizan una jeringuilla calefactada a modo de extrusión. El extrusor convencional se compone principalmente de:

- Motor paso a paso para hacer que el filamento avance hacia la boquilla.
- Resistencia calefactora que sirve para derretir el plástico del filamento.
- Boquilla por la que se dispensa el plástico ya derretido. Hay de varios diámetros(desde 0.2 a 1 mm). Cuanto menor sea el diámetro de la boquilla, más precisa será la impresión pero más lenta.

Algunos modelos, como el mk8, incluyen un sistema de refrigeración sencillo para evitar sobrecalentamiento. En la figura 2.7 podemos ver las partes que componen un extrusor, concretamente el extrusor mk8 mencionado anteriormente. Además, la resistencia suele ir protegida por una capa de teflón u otros materiales para evitar quemaduras o posibles problemas. Recordemos que ésta debe alcanzar una temperatura aproximadamente de 300°, lo cual es bastante peligroso y si no se controla podría haber algún accidente.

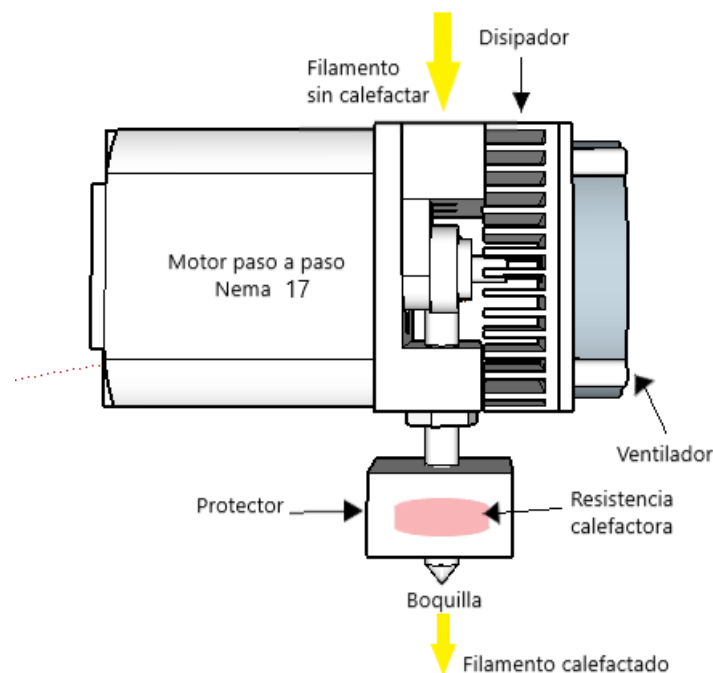


Figura 2.7: Extrusor o HotEnd: modelo mk8.

2.4.5. Controladores

Para controlar los motores se usan unos elementos llamados variadores, los cuales hacen de interfaz entre el programa que gobierna el movimiento del motor y el motor en sí. En general, en estos variadores vienen incorporados unos sistemas de control para saber cuanto ha rotado el motor, lo cual sirve para tener controlados los finales de carrera, y sistemas para controlar que la temperatura de los motores no exceda los límites de funcionamiento normal, apagado automático del variador cuando detecta sobre calentamiento. Aun así, existen sistemas complementarios a los variadores que, en el caso de ciertos variadores no muy potentes, ayudan a controlar los finales de carrera. Para controlar los finales de carrera hay dos técnicas:

- **Mecánica:** Un mecanismo mecánico advierte al programa de que se ha llegado al final de carrera. Por ejemplo un pulsador colocado al final de un eje, encoders, etc.
- **Digital:** Llevar una cuenta de cuanto se ha avanzado calculando el avance por tiempo. Estos métodos en general suelen ser muy poco recomendados o en su defecto se recomienda que si se usan se basen en técnicas mecánicas ya que el grado de error que se puede cometer es alto a medida que se avanza en la impresión.

Para controlar la temperatura se usan termistores de alta resistencia que advierten de la temperatura alcanzada por cada motor o por el extrusor de modo que nunca exceda el límite de seguridad.

En general, la clave de un buen control de los motores reside en controlar su temperatura y su posición en todo momento. El cómo llevar a cabo este control depende completamente del tipo de motor que usemos. En este caso los diferenciaremos en dos clases:

2.4.5.1. Motores de continua

Estos, a diferencia de los motores bipolares, no tienen ninguna garantía de control de movimiento, es por esta razón que de implementarse para el uso de impresoras requerirían de los mecanismos mencionados anteriormente para controlar los finales de carrera y las temperaturas.

El funcionamiento de los motores de continua se basa en la repulsión que se ejerce entre los polos magnéticos del imán fijo del motor y un electroimán que está dentro del motor. El electroimán está montado sobre un rotor y su eje le permite rotar entre. De modo que, cuando la corriente pasa por el electroimán se genera un campo electromagnético que interactúa con los imanes fijos del motor, rompiendo la inercia del rotor y haciendo que este comience a girar. El sentido del giro depende de cómo se encuentre conectada al circuito la fuente de alimentación del motor [14].

Ante la presencia del campo magnético el rotor gira sobre sí mismo.

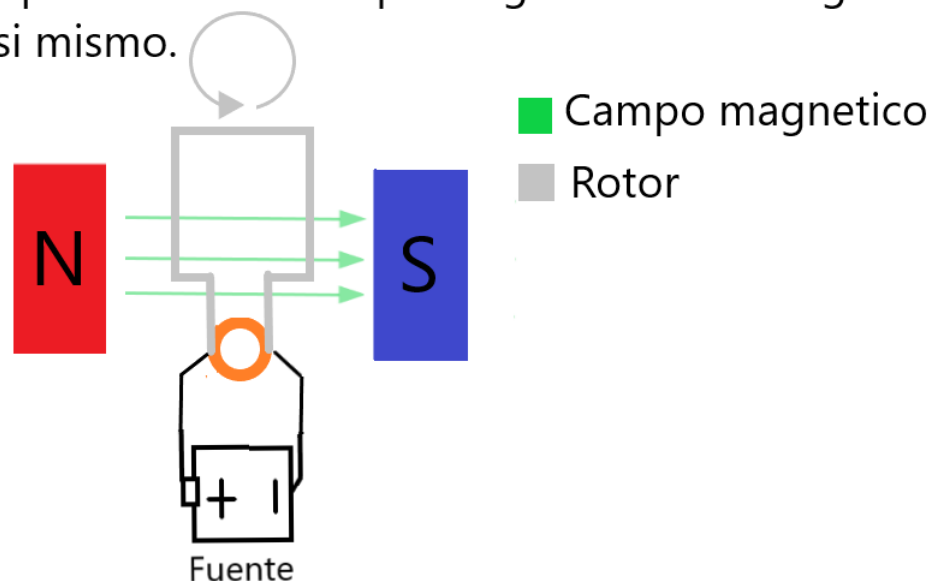


Figura 2.8: Esquema de funcionamiento de un motor de continua.

2.4.5.2. Motores bipolares

Estos motores, junto con un variador adecuado y potente, no requerirían de los sistemas de control externos mencionados anteriormente. Estos tienen, de manera nativa, unos mecanismos mecánico-magnéticos de avance muy precisos que garantizan que si el programa que controla este motor le ordena avanzar una distancia concreta, estos con una precisión muy alta lo harán.

Este tipo de motores son los que se usarán en la implementación de la impresora ya que son idóneos para tener un control de movimiento y temperatura de la impresora mejores. En la sección 4.2.2 se explicarán con más detalle, así como el modelo de motores usados.

2.4.6. Placa

La placa es un sistema hardware que ejecuta las instrucciones pertinentes para llevar a cabo la impresión. Estas deben encargarse de ejecutar toda la lógica que hay detrás del proceso de impresión y por lo general, no tienen integrados todos los componentes hardware necesarios para controlar la impresora, es por ello que se suele montar sobre estas placas controladoras otras placas adicionales que si poseen estos elementos. Se pueden usar gran variedad de placas controladoras como es el caso de Raspberry pi, UP Core, C.H.I.P Pro Dev Kit, Orange Pi Plus 2E, Arduino, etc. En el caso de este proyecto la decisión de que placa usar esta entre Arduino o Raspberry, ya que son placas típicas de desarrollo fáciles de entender y de modificar, son muy económicas y además cuentan con grandes comunidades que desarrollan para estas plataformas.

A continuación se expondrán las ventajas e inconvenientes de usar Arduino.

Ventajas:

- Tiene una comunidad muy activa y extensa en la cual hay mucho software y tutoriales sobre la aplicación de esta plataforma al mundo de la impresión 3D.
- El lenguaje es sencillo y fácil de aprender.
- Al ser un sistema dedicado al contrario que un PC, se ejecuta un solo programa y éste no se ve interrumpido o expropiado por el planificador de tareas como en otros sistemas que sí albergan un sistema operativo, lo cual es idóneo para no interrumpir el proceso de impresión y que éste no tenga fallos.

Inconvenientes:

- Arduino solo tiene, de manera nativa, su propio lenguaje de programación el cual esta basado en c y es poco universal y poco potente. Se podría preparar software para que pudiera compilar otros lenguajes pero esto requeriría una inversión de tiempo considerable.

- Implementar la impresora con Arduino implica que debe ser un sistema autónomo no dependiente de otros elementos, lo cual deja a la placa hacer todo el trabajo cuando en realidad se podría repartir la carga y hacer que fuera más rápido gracias a la concurrencia de threads entre diferentes hilos de diferentes máquinas. Existen librerías de usuarios que emulan este comportamiento, pero no es real y no tiene toda la potencia de otras máquinas que si están pensadas para ello.

- Usar Arduino implicaría cerrar la puerta a posibles modificaciones futuras de manera más sencilla. Es decir, si quisiéramos añadir nuevas funcionalidades adicionales a la placa, necesitaríamos hardware de control adicional, en cambio, con una máquina que albergue un sistema operativo este proceso de ampliar funcionalidades sería más sencillo.

Es por este balance entre ventajas y desventajas del uso de Arduino como placa de la impresora lo que provocó que se descartara y en su lugar se usara Raspberry pi ya que éste, además de haber sido tratado durante la carrera y que esto me confiere una mayor familiaridad con éste sistema, permite procesamiento concurrente de tareas y además, es un hardware que, aunque muy básico, comparado con otras placas de control que hay en el mercado goza de una popularidad muy grande lo cual hace que tenga una comunidad muy activa que desarrolla mucho software para esta plataforma. Además, al ser una placa bastante sencilla es fácil de entender para usuarios poco experimentados, lo cual hace que sea perfecto para los propósitos de este proyecto que, si se recuerda, es el de crear un punto de partida fácil y sencillo de acceder para que los usuarios puedan aportar sus ideas, desarrollos e implementaciones a este proyecto.

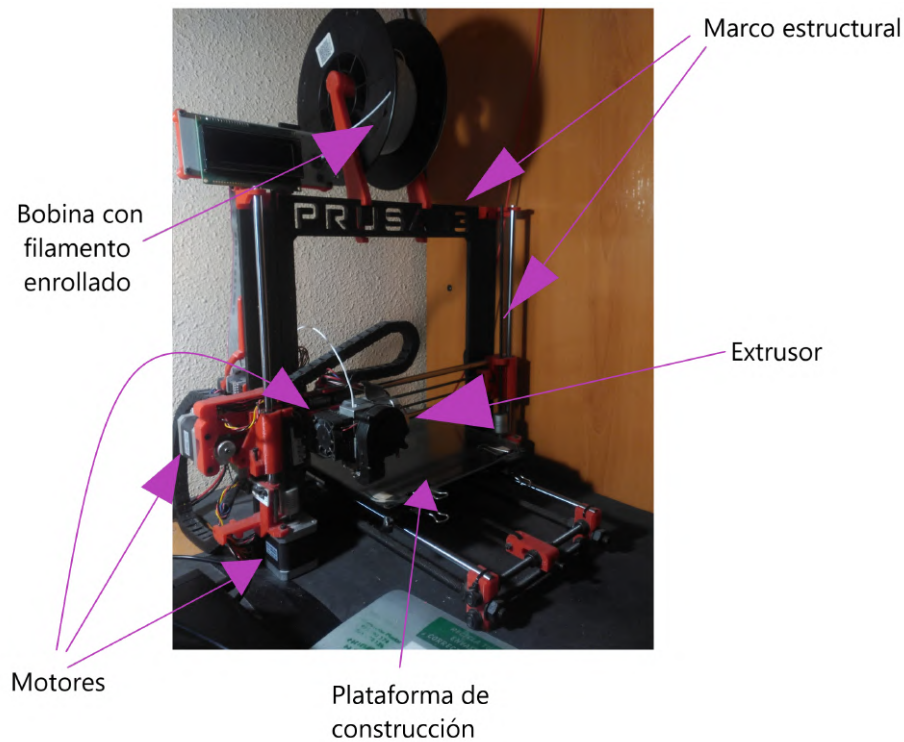


Figura 2.9: Partes de una impresora 3D **FDM** : Modelo PRUSA.

2.5. Proceso de Impresión

El proceso general para obtener una pieza impresa requiere de tres pasos:

- Obtención del **modelo 3D** de la pieza.
- Traducción del modelo a un lenguaje legible por la placa.
- Impresión de la pieza.

2.5.1. Diseño 3D

Para obtener el **modelo 3D** de una pieza se puede hacer por dos métodos:

- Modelándola con alguna aplicación.
- Escaneado tridimensional del objeto real.

2.5.1.1. Modelado 3D

Hay abundante softwares, tutoriales y comunidades en la red que hablan acerca del modelado de figuras con aplicaciones y explican técnicas y usos para mejorar el estilo, la calidad y la variedad de formas a describir en nuestro modelo. Hay tanto aplicaciones gratuitas como de pago.

2.5.1.2. Escaneo tridimensional

Este método permite obtener modelos digitales 3D a partir del análisis de la forma y el color del objeto a reproducir. El proceso se inicia mediante un haz de luz láser proyectado sobre el objeto. El haz rebota en la superficie del objeto y vuelve a la fuente de emisión. Esto permite a la placa hacer un mapa de puntos o nube de puntos tridimensionales que representan la figura. Posteriormente debe pasar por algún software que simplifique esta nube de puntos de un cardinal alto a una serie de triángulos que simplifique la superficie de la figura. El algoritmo que utilizan estos softwares de simplificación de superficies se llama Triangulación Voraz y están basados en la triangulación de Delaunay [51] [49].

El algoritmo de Triangulación Voraz sirve para calcular la triangulación de una figura tridimensional, representada por un polígono o por una nube de puntos. La estrategia que sigue el algoritmo consiste en unir aristas a la solución, si la nueva arista a añadir no intersecta con ninguna otra previamente añadida, uniendo el par de vértices más próximos entre si.

La estrategia más sencilla para implementar este algoritmo es la siguiente:

- 1 Crear un solución inicial.
- 2 Obtener una lista de aristas prometedoras, combinando los vértices de en pares.
- 3 Ordenar la lista de aristas prometedoras de menor a mayor longitud.
- 4 Mientras la lista de aristas prometedores tenga alguna arista se van comprobando los cruces entre aristas.
 - 4.1 Se obtiene la primera arista de la lista y se saca de la lista de aristas prometedoras.
 - 4.2 Si la arista en cuestión no se cruza con ninguna otra, la consideramos como válida y formará parte de la solución.

Este algoritmo también es usado por las aplicaciones de modelado 3D para convertir los modelos hechos a base de puntos en triángulos que describen la superficie. En la figura 2.10 se puede ver un ejemplo de la triangulación.

La idea que subyace detrás de la aplicación de este algoritmo es sencilla. La nube de puntos que representa un objeto suele tener un cardinal muy elevado y extrapolar de ella datos fundamentales como cuales son las caras superficiales, interiores y demás detalles es bastante arduo. Pero gracias al físico-matemático Delaunay se averiguó que no eran necesaria tal nube de puntos para representar un objeto tridimensional, solo bastaba un número de triángulos, dramáticamente menor al cardinal de puntos de la nube de puntos, para llevar a cabo esta tarea.

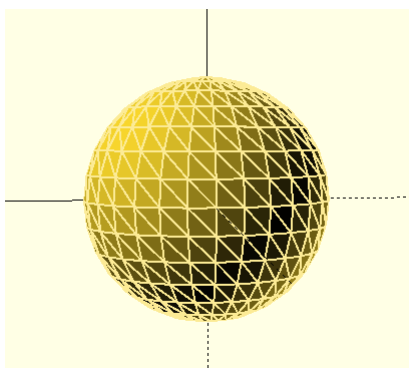


Figura 2.10: Ejemplo de triangulación de nube de puntos de una esfera en OpenSCAD.

2.5.2. Traducción del modelo

Una vez obtenido el modelo tridimensional, la siguiente etapa consiste en transmitirle la información característica de la figura a la placa controladora para que ésta sepa si puede o no imprimirla y en caso afirmativo llevarla a cabo. En general la información de la figura viene contenida en un archivo de texto que contiene el número de triángulos mínimo para representar la figura, los 3 puntos tridimensionales que conforma cada triángulo y a qué cara pertenece dicho triángulo (esto se suele definir por la normal). Este formato se denomina .STL. En la figura 2.11 viene un ejemplo donde se explica su estructura. Cada software implementa el formato de las figuras de una manera y este es uno de los motivos por los que no todas las impresoras pueden entender todos los formatos.

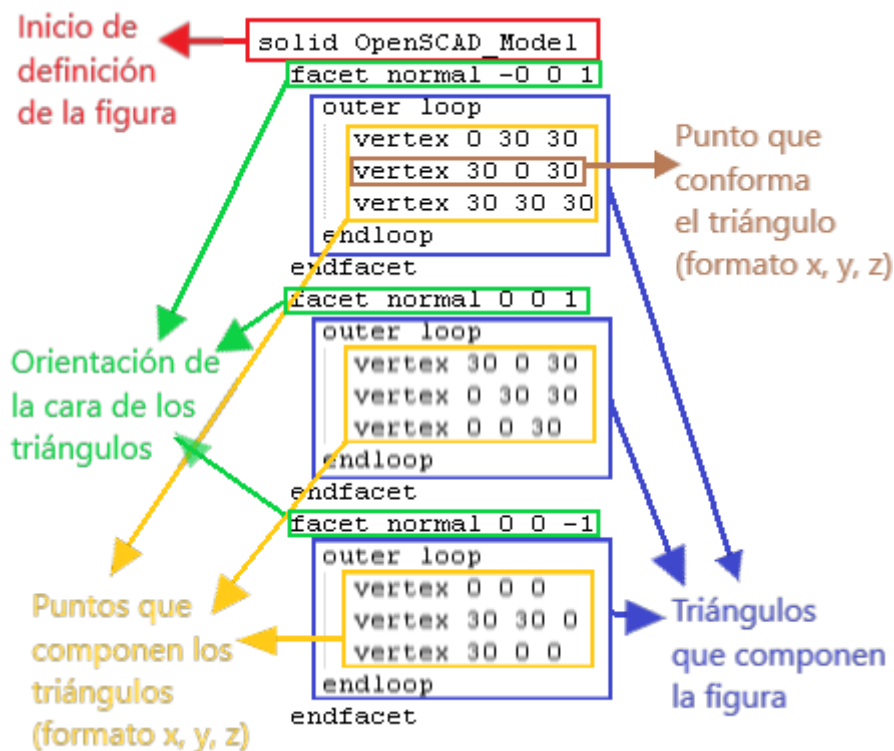


Figura 2.11: Ejemplo de archivo de texto que contiene la información de una figura en formato .STL: OpenSCAD.

Gracias a este archivo de entrada, el algoritmo que opera en la impresora 3D es capaz de obtener las coordenadas por donde debe pasar el extrusor depositando el material fundido capa a capa. El algoritmo debe rebanar la figura en múltiples capas en el sentido del eje Z. Para cada una de las secciones formadas por las capas se deposita el polímero fundido que ocupa completamente un área. Una vez terminada la primera sección, y al mismo tiempo que se va solidificando, la boquilla extrusora asciende en la altura Z y comienza a imprimir la siguiente sección y así sucesivamente. En la figura 2.12 se muestra el proceso de rebanamiento, capa a capa, de una figura a imprimir.

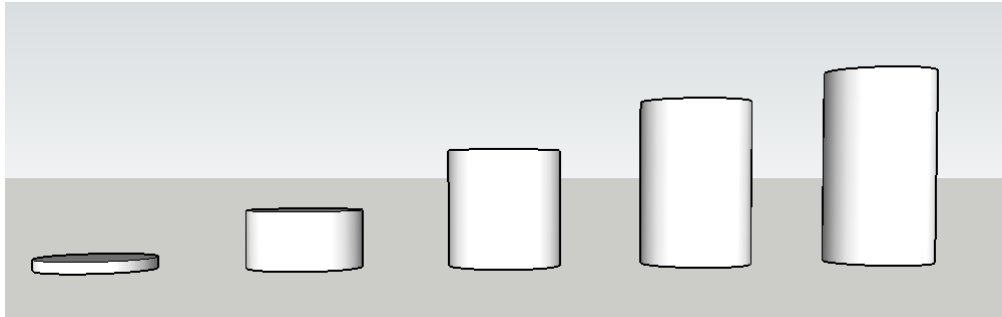


Figura 2.12: Rebanamiento de figura en sentido del eje Z.

Una vez obtenidas esas rebanadas, obtiene las trayectorias que debe seguir el extrusor para imprimir la figura.

Existe un lenguaje de programación llamado G-Code con el que se puede implementar el algoritmo de recorrido de trayectorias. Es cual es el más estandarizado y fue desarrollado por el MIT en la década de los 50.

El problema que plantea el G-code es que es dependiente de cada máquina. Eso quiere decir que como cada máquina tiene una serie de motores colocados en distintos sitios y que accionan los movimientos de distintas maneras, tiene como consecuencia lógica que cada máquina tenga su propia distribución del lenguaje G-code.

2.5.3. Impresión

Una vez que la placa de control ya conoce las características de la figura y sabe que trayectorias debe seguir para obtenerla, ésta se carga en su memoria y comienza la impresión.

Cuanto más detalles tenga la figura y/o grande sea, más trayectorias tendrá que seguir el extrusor y más lenta será la impresión. En general, para evitar el mal gasto de material o pérdidas de tiempo innecesarias, se suele hacer que la pieza no sea completamente llena por dentro, sino que se imprimen estructuras de soporte internas para asegurar la estabilidad mecánica y estructural de la figura.

2.6. La impresión 3D ahora

Las aplicaciones 3D son múltiples y cada día aparecen nuevos campos de aplicación. Algunos de ellos son los sectores de alimentación y farmacia, aeronáutico, automoción, deportes,

medicina, moda y joyería, construcción, etc. A continuación se enumeran algunos de los avances de la impresión 3D realizados en los últimos años.

2.6.1. Alimentación y farmacia

La agencia Espacial Americana (NASA) trabaja junto a la empresa *Systems and Materials Research Corporation* (SMRC) en la fabricación de una impresora 3D para la fabricar alimentos en el espacio para los futuros viajes tripulados [15].

La empresa española *Natural Machines* [16] ha desarrollado una impresora 3D de comida. La impresora funciona con conexión a internet y a partir de una pantalla táctil se pueden elegir recetas o diseñar un plato original.

En el mercado se pueden encontrar otras impresoras de alimentos como la *F3D*, la *Chefft* o la *Candy* fabricadas por 3D Systems [17].

La FDA (*U.S. Food and Drug Administration*) ha aprobado la producción por impresión 3D de un medicamento para la epilepsia, el Spritam. La impresión capa a capa del polvo del medicamento facilita no solo la posibilidad de ajustar la dosis en función del paciente, sino su disolución después de ser ingerida debido al proceso de fabricación de esta pastilla [18].

2.6.2. Ámbito domestico

El movimiento **RepRap**, junto a la aparición de **MakerFaire** de impresión 3D, ha permitido compartir el conocimiento de la tecnología de la impresión 3D **FDM**. Actualmente se pueden adquirir impresoras domésticas de baja calidad por 300 dólares [1], e incluso a partir de kits de montaje y software gratuito por un precio inferior aunque de una calidad baja.

Nike [19] anuncia que en un futuro sus clientes podrán imprimir en 3D sus zapatillas en su propia casa y según sus propias preferencias.

Uno de los principales usos de las impresoras 3D en el ámbito domestico es la impresión de piezas de repuestos después de su rediseño o escaneo 3D. En repositorios como *Thingiverse* se pueden encontrar piezas de repuestos. Actualmente cuenta con más de treinta mil **modelos 3D** y ya se han descargado unos diez millones de archivos.

2.6.3. Aeronáutica

Airbus [20] está empezando a dar forma al futuro de la fabricación de componentes de aeronaves para sus aviones. En el modelo *Airbus A350 XWB*, con más de seis millones y medio de piezas, hay más de seis mil quinientas que se pueden fabricar por impresión 3D. El objetivo es reducir el peso y el coste final de la pieza evitando la fabricación de moldes y utillajes, la mecanización o los cortes.

La *Agencia Espacial Europea* [21] ha diseñado una impresora POP3D para la Estación Espacial Internacional la cual tiene unas dimensiones de 25 x 25 x 25 cm y 5,5 kg. Es bastante compacta y es capaz de imprimir en plástico biodegradable de tipo PLA con **FDM**.

La *Agencia Espacial Americana, NASA* [22] está trabajando en el proyecto de impresión 3D en gravedad cero. Los estudios del proyecto han demostrado que la impresora 3D funciona con normalidad en el espacio a pesar de la baja temperatura y la ausencia de gravedad. Sin duda, este es el primer paso de cara a futuras misiones tripuladas. También tienen otro proyecto en el que consiguieron imprimir un prototipo de un motor de cohete con una potencia superior a nueve toneladas.

Boeing [23] y *GE Aviation* [24] utilizan impresoras 3D para fabricar algunas de las piezas de los motores a reacción más grandes y potentes del mundo (GE90-94B), inyectores de combustibles y otras piezas para el motor GE9X de los nuevos modelos del 777X de Boeing. El GE9X está considerado el mayor motor a reacción construido.

2.6.4. Automoción

La escudería de fórmula uno *Williams* [25] colabora con *EOS* [26] en el uso de las tecnologías de fabricación aditiva para la fabricación avanzada de piezas para sus monoplazas.

La empresa *Local Motors* [27] ha desarrollado uno de los primeros prototipos de coches impresos 3D, llamado *LM3D Swim*. El precio final es de cincuenta mil euros y prácticamente el ochenta por ciento se imprime con diferentes tipos de polímeros y fibra de carbono. Su impresión se realiza en micro fábricas que la empresa tiene distribuidas por el mundo.

2.6.5. Deportes

Adidas, *Nike* y *New Balance* diseñan suelas de calzado deportivo para ser impresas en 3D con materiales poliméricos capaces de absorber el impacto de la pisada. En el caso de *New Balance*, su alianza con *Nervous Systems* [28] ha permitido diseñar suelas con porosidad variable en distintas regiones de la suela.

El equipo de esgrima japonés ganó la medalla de plata en los juegos olímpicos de 2012 con espadas cuyas empuñaduras se imprimieron en 3D. La impresión se realizó en la universidad de *Tsukuba* con la impresora 3D *Objec350 Connex* de la empresa *Stratasys* [29].

2.6.6. Medicina y biología

La empresa *Materialise* [30] y la *Universidad de Michigan* [31] firmaron un acuerdo de colaboración para diseñar férulas traqueales por impresión 3D adaptadas a los pacientes.

El centro de investigación *Lawrence Livermore National Laboratory* de los Estados Unidos de América, trabaja en la impresión de tejido humano con vasos sanguíneos mediante bioimpresión 3D compatible con el cuerpo humano.

En la universidad *Wake Forest* [32] han conseguido imprimir en 3D piel. La piel obtenida permite facilitar la cicatrización de heridas y quemaduras.

La empresa *Organovo* [33] anunció que habían logrado imprimir capas de hígado mediante las nuevas tecnologías. Las impresoras pueden coger células y depositarlas sobre un andamio poroso para formar el órgano a medida del paciente. Con esta tecnología ya se ha conseguido crear y trasplantar una tráquea utilizando las células madre del paciente. También han conseguido imprimir vasos sanguíneos y tejidos. La revista *Time* la ha considerado una de las cincuenta empresas más innovadoras del mundo.

El departamento de Ingeniería Mecánica del *Instituto Tecnológico de Canarias* [34] imprime prótesis personalizadas de titanio poroso. También ha diseñado y fabricado las primeras prótesis a medida para reconstrucciones óseas faciales.

La empresa *Podoactiva* [35] diseña y fabrica máscaras de protección facial personalizadas mediante impresión 3D. El proceso empieza con la reconstrucción tridimensional de la cara del paciente a partir del escaneo 3D y finaliza con la impresión del molde. Las máscaras se refuerzan con fibra de carbono.

La ONG *Magic Arms* [36] fabrica exoesqueletos de brazos para niños.

La comunidad *e-nable* [37] ha diseñado una colección de prótesis imprimibles 3D que se pueden definir a partir de las medidas antropométricas de los pacientes. Su funcionamiento es sencillo, solo se debe seleccionar el modelo de prótesis, introducir sus medidas y descargar los archivos para su posterior impresión.

Capítulo 3

Fase uno: Diseño estructural

3.1. I+D previo

Antes de comenzar a hacer el diseño definitivo de la impresora, se procedió a desarrollar una serie de pre-prototipos.

Los objetivos de la fase previa de diseño eran:

- **Adquirir conocimientos de las mecánicas** de movimientos a los que debía enfrentarse la impresora.
- **Encontrar una estructura sólida de soporte**, de manera experimental, que fuera económicamente viable, fácil de construir y robusta.
- **Familiarizarse con los problemas estructurales** más comunes.
- **Maximizar la precisión y la velocidad** de impresión en la medida de lo posible.

Podemos dividir esta etapa de I+D en tres subetapas. En cada una de ellas se obtuvo un prototipo de impresora y unas conclusiones acerca del pre-prototipo que se estaba construyendo en cada etapa. Estas conclusiones servirían como punto de partida para el prototipo definitivo. Todas ellas estaban hechas de madera, material muy poco recomendable para la versión definitiva de la impresora. El motivo por el cual se hicieron las piezas estructurales de los pre-prototipos de madera es que era más sencillo de perfilar, cortar y modelar las piezas para hacer pruebas que si fuesen de otros materiales como metal u otros. También influyó el hecho de que era más económico hacer pre-prototipos de madera.

A continuación se expondrán las tres subetapas en cuestión.

3.1.1. Primera versión

Ésta primera versión fue plenamente experimental poniendo en práctica ciertas ideas teóricas previas de como podría ser la estructura. Se trataba de obtener unos conocimientos básicos que cimentaran todo el proceso.

Las conclusiones adquiridas una vez alcanzado un prototipo funcional y operativo fueron:

- La precisión lo es todo. La imprecisión derivada de la baja calidad de las herramientas y de la pericia de quien las maneja generaba prototipos poco eficientes. Este aspecto debía ser paliado. Había que comprar mejor equipo y mejorar las técnicas de construcción y modificación de las piezas.
- Las uniones metálicas debían ser reforzadas o llevadas a cabo por enganches, no por soldaduras. Dan muy mal resultado sin el material adecuado para hacerlas.
- Los motores Nema 17 tienen gran fuerza de empuje o torque, pero no infinita. Había que aligerar la carga estructural o carga estática lo máximo posible.
- La calidad del material influye notablemente en la precisión. Además controlar la fricción resistente al desplazamiento es vital, frena mucho más de lo esperado el movimiento.
- Cuanto menor paso por vuelta tuviera el husillo o la varilla roscada tendría menor avance por paso de motor pero mayor precisión.

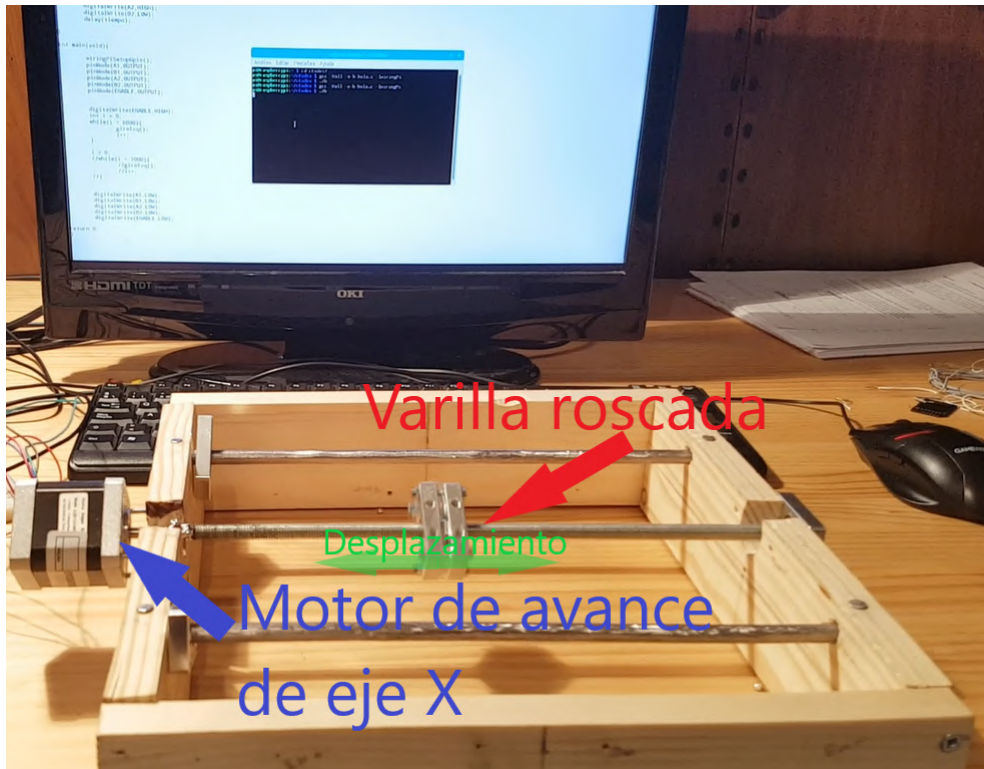


Figura 3.1: Primera versión de la impresora. Foto tomada durante la prueba de funcionalidad.

3.1.2. Segunda versión

Una vez analizadas las conclusiones obtenidas de la primera versión comenzó la segunda subetapa. En esta se trata de recolectar la mayor información posible y simultáneamente crear una estructura funcional mejor que la anterior.



Figura 3.2: Segunda versión de la impresora. Foto tomada durante la prueba de rendimiento y funcionalidad.

Las conclusiones adquiridas una vez alcanzado un prototipo funcional y operativo fueron:

- El aumento de tamaño provoca que sea fácil cometer imprecisiones a la hora de fabricar las piezas.
- Es importante que la base sea poco pesada, para facilitar el transporte.
- La madera de conglomerado es de baja calidad y la calidad de los cortes es pobre, lo cual produce errores extra que no deben ser considerados como estructurales per se.
- La varilla roscada no debe tener excesivos pasos por vuelta, sino se ralentiza mucho la velocidad.
- Es importante guiar el eje X con carriles para evitar errores de desviaciones.
- Las vibraciones son un problema a tener en cuenta, los motores hacen vibrar la estructura y esto genera imprecisiones.

3.1.3. Tercera versión

En este prototipo se buscaba profundizar aun más en los conocimientos adquiridos anteriormente, eliminando las variables (como el tipo de material usado) de la ecuación para que no afectarán en la evaluación estructural de la impresora ya que en el futuro se pretendía hacer la estructura de otro material distinto a la madera.

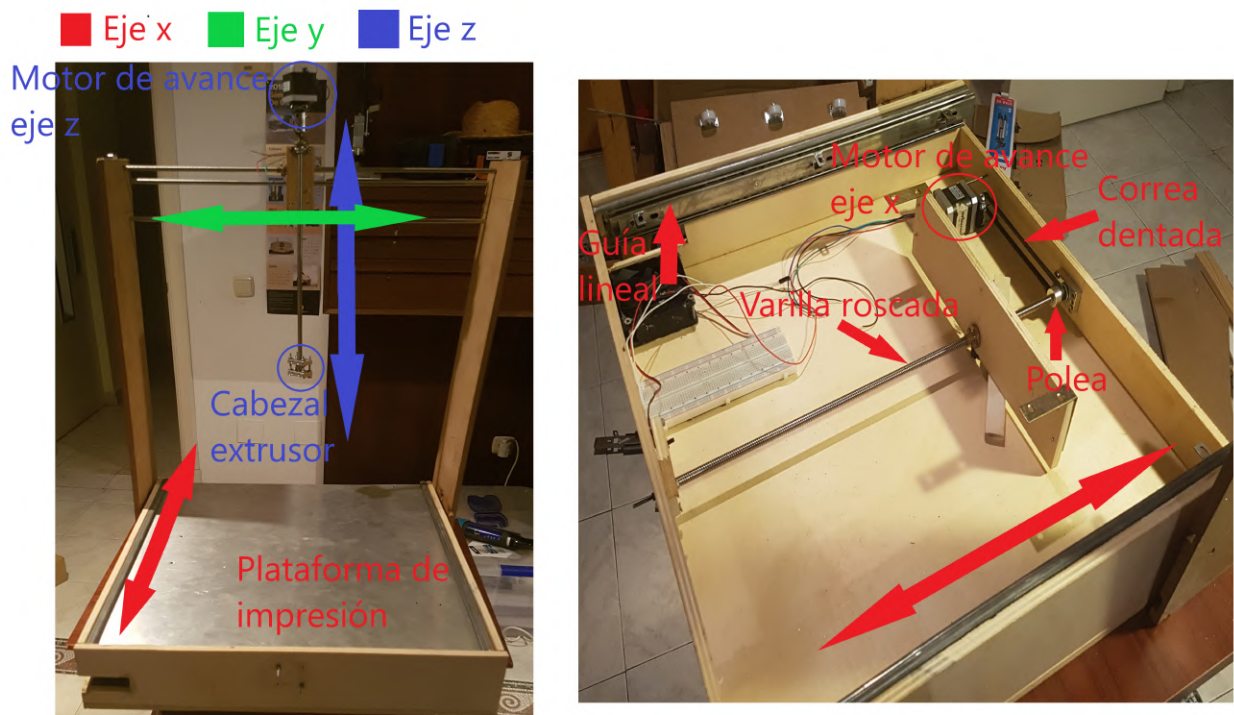


Figura 3.3: Tercera versión de la impresora.

Las conclusiones adquiridas una vez alcanzado un prototipo funcional y operativo fueron:

- El contrachapado es una buena opción para construir la base, el material que se use para la versión definitiva debe tener propiedades similares de estabilidad y superiores en cuanto a la dureza.
- El soporte del eje Y y consecuentemente el Z debe ser mucho más sólido. La estructura lo aguanta pero las vibraciones y el pandeo son notables. Hay que buscar otra estructura más robusta.

- No hay que desatender el cableado de la impresora. Esto puede ser un factor decisivo a la hora de que la estructura sea viable o no. También hay que tener en cuenta donde debe ir colocada la controladora de la impresora. La ventilación de ésta es vital para evitar que se produzca un fallo por sobrecalentamiento. Colocarla debajo de la plataforma de extrusión no es recomendable.
- La precisión manual no es suficiente para alcanzar un nivel aceptable de calidad de obtención de piezas. Será necesario buscar alternativas a la manufacturación casera de las piezas que componen la impresora.
- El calor es un factor a tener en cuenta. Los materiales que compongan la estructura deben ser ignífugos y tolerar bien el calor. Además la estructura debe tener elementos de apoyo para ayudar a disipar el calor generado dentro del interior de la base.
- El sistema de poleas es bueno. Antes se hacía rotar directamente las varillas con unas uniones metálicas entre los motores y éstas, pero el problema es que en los cambio bruscos de direcciones sobre los ejes X e Y hacían que la estructura sufriera vibraciones. Como lado negativo, el sistema de poleas resta velocidad al movimiento.

Para probar que las estructuras funcionaban, se diseñó un programa sencillo de control de los tres motores que controlaban los ejes para desplazar la plataforma y el extrusor. El programa recibía como input un carácter hacia la dirección en la que se quería que se desplazara la impresora y éste lo ejecutaba en función de veces que se había introducido ese carácter.

Tras las indagaciones recabadas durante esta fase de I+D previo se hizo una síntesis de todas estas y se concluyó lo siguiente:

- El aluminio sería el material idóneo, ya que es fácil de malear, cortar y tiene propiedades térmicas y mecánicas muy aceptables.
- La estructura debe ser muy sólida, más de lo esperado. La estructura debe ser capaz de paliar las vibraciones producidas por los motores, que al ser cuatro, producen una vibración considerable. También debe resistir los cambios de dirección en los ejes. Dado que cualquier cuerpo tiende a seguir su trayectoria por su propia inercia, el cambio brusco de dirección provoca una sacudida importante. Hay que reducir la masa del cabezal y hacer que la estructura sea más robusta para reducir al máximo la repercusión de este fenómeno.

- En todas las versiones de pre-prototipos el eje x debía transportar una plataforma y consecuentemente la pieza que se estaba imprimiendo. Esto es una mala idea ya que hace que el eje x tenga un límite de transporte, dependiente del peso de la figura. Eso quiere decir que si la figura es lo suficientemente pesada, hará que el eje x no pueda desplazarse. Esto viola el principio de escalabilidad que todo buen diseño, de cualquier ámbito, debe perseguir.

Una vez obtenido el suficiente conocimiento y las bases para construir una estructura que atendiera a las necesidades y solventara los problemas percibidos se procedió al diseño definitivo del prototipo.

3.2. Prototipo definitivo

Después de un tiempo, tras haber madurado todos los conocimientos adquiridos en la fase anterior y haciendo ciertos calculo básicos (Estos se hicieron de manera orientativa, basándose en un libro llamado Números Gordos [52]) se llegó a la version actual.

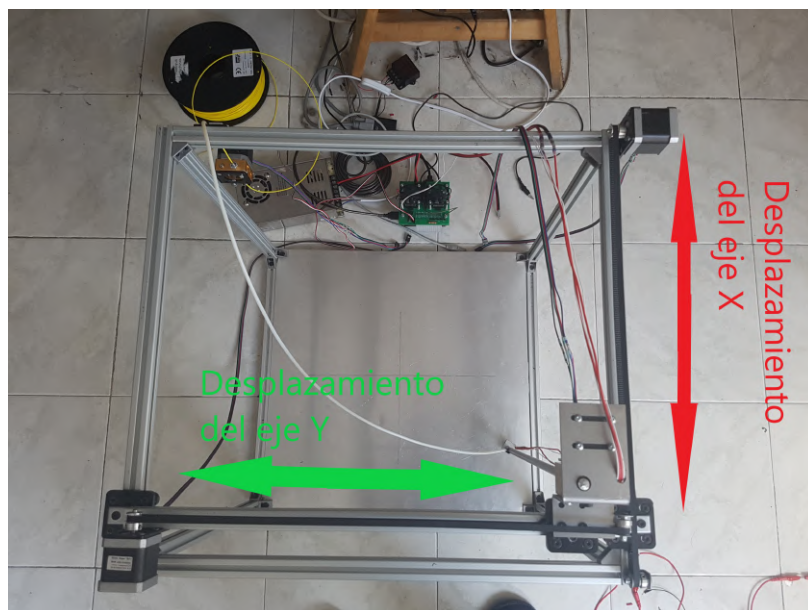


Figura 3.4: Vista de planta de la impresora.

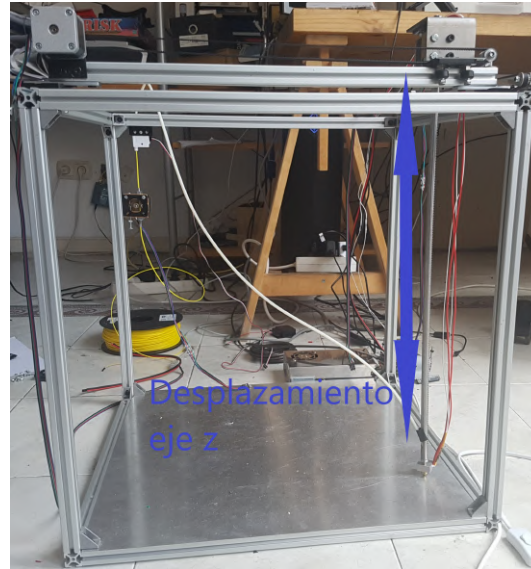


Figura 3.5: Vista alzada de la impresora.

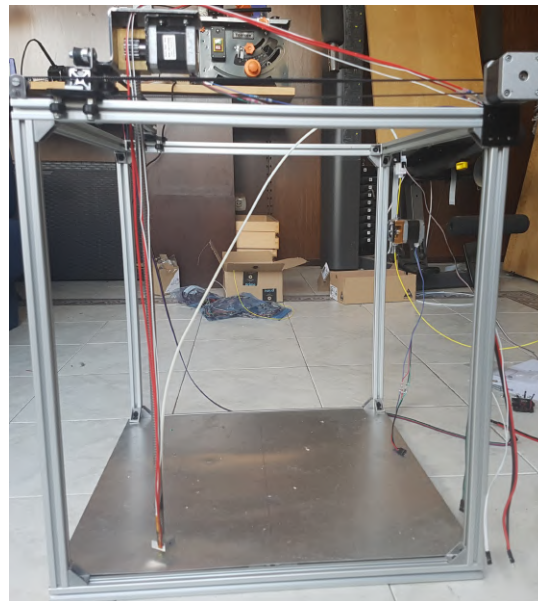


Figura 3.6: Vista de perfil la impresora.

3.2.1. Aplicación del conocimiento adquirido en la fase de I+D

A continuación se expone como, siguiendo las lecciones aprendidas en la fase de I+D previo, se han paliado los errores encontrados a los que se enfrenta la estructura.

- Toda la estructura esta hecha con aluminio, con perfiles que confieren a la estructura una

gran robustez, haciéndola aguantar y frenar considerablemente las vibraciones y contoneos producidos por los cambios bruscos de dirección en los ejes.

- El sistema de movimiento esta basado en poleas, que mejoran notablemente el control del movimiento y cambios bruscos de dirección.
- Se ha desligado el eje x de la plataforma de soporte, de modo que la impresora es escalable en tamaño.
- Se han eliminado los husillos como forma mecánica de avance en los ejes, en su lugar se han usado patines, engranajes y cremalleras dentadas, lo cual hace que el movimiento sea más suave y menos resistivo al rozamiento de los materiales.
- Se ha aligerado la carga del cabezal extrusor, reduciendo considerablemente el peso de éste y haciendo que los contoneos producidos por los cambios bruscos de dirección en los ejes sean menos severos. Para reducir el peso del cabezal extrusor se ha distribuido la carga del mk8. El motor ha pasado a estar montado sobre uno de los perfiles de la estructura y el cabezal calefactor ahora va insertado en el cabezal del eje z.
- La controladora de la impresora ya no está dentro de la propia estructura de la impresora para evitar que el calor del plástico fundido (recordemos que alcanza temperaturas cercanas a los 300°) sobrecaliente la controladora y en general el hardware.

3.2.2. Partes de la impresora y lista de materiales

3.2.2.1. Vistas acotadas del marco estructural

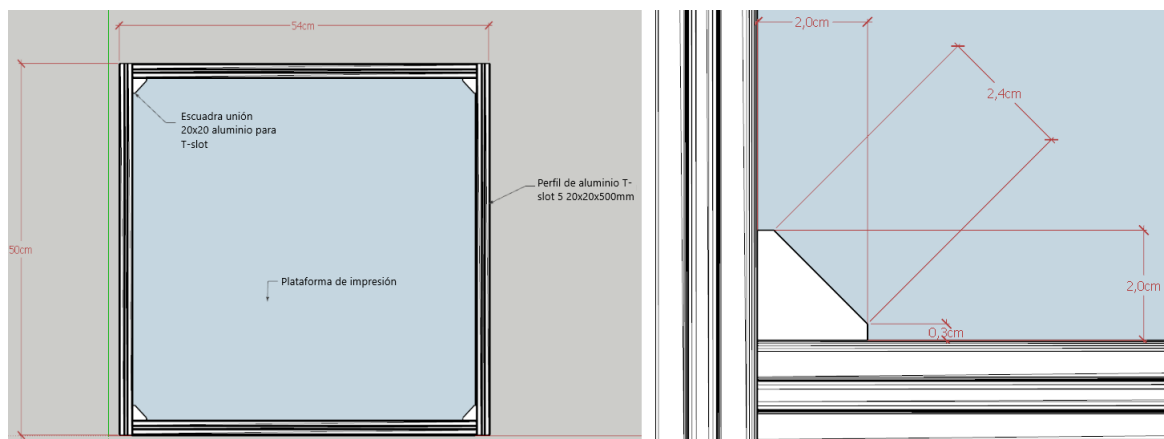


Figura 3.7: Vista acotada desde arriba del marco estructural.

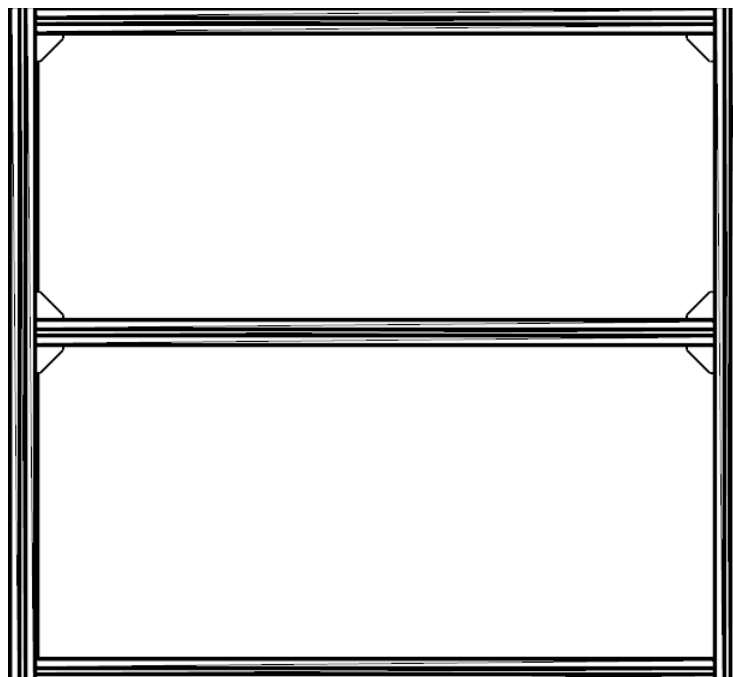


Figura 3.8: Vista desde abajo del marco estructural.

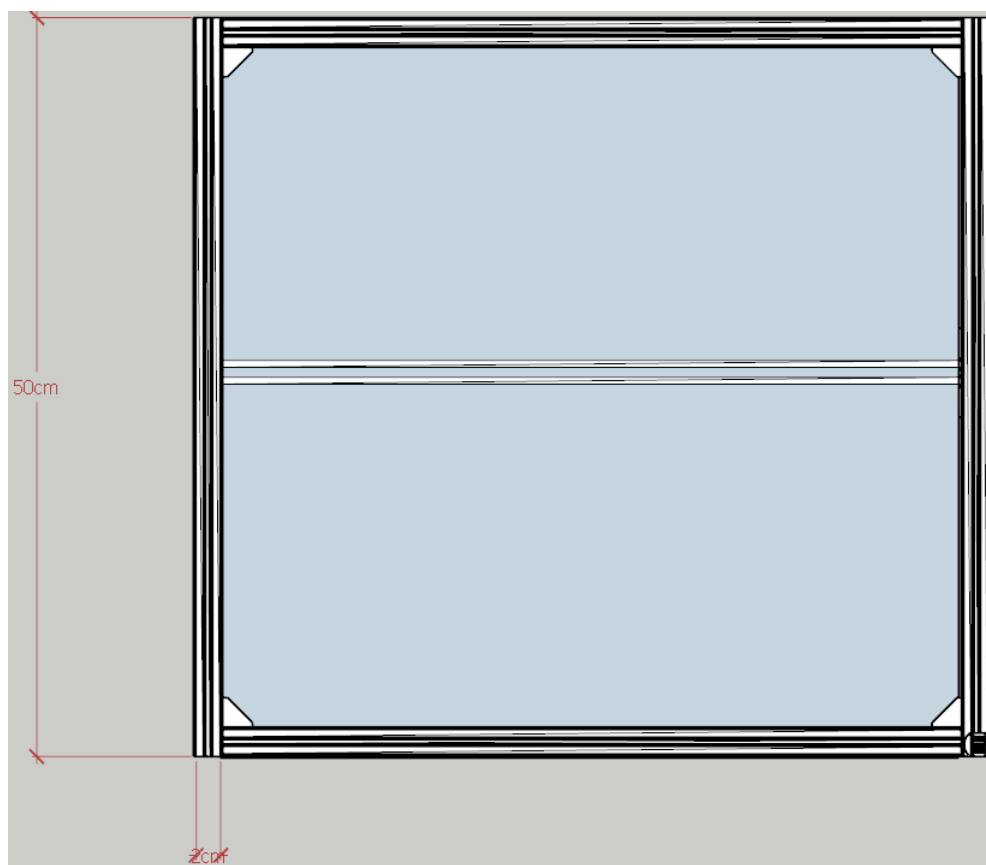


Figura 3.9: Vista acotada frontal del marco estructural.

3.2.2.2. Vistas acotadas del eje X

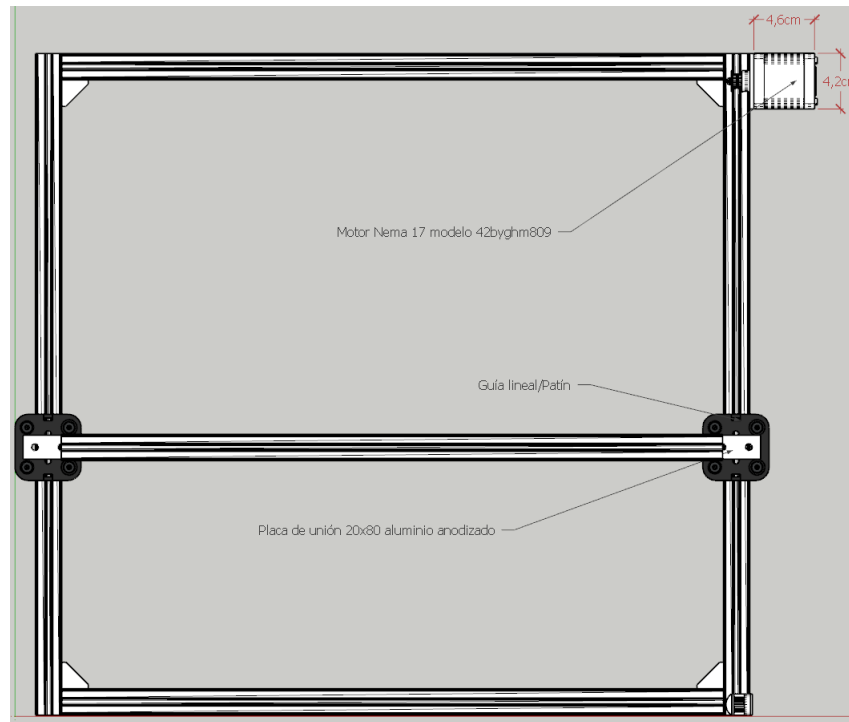


Figura 3.10: Primera vista acotada desde arriba del eje X.

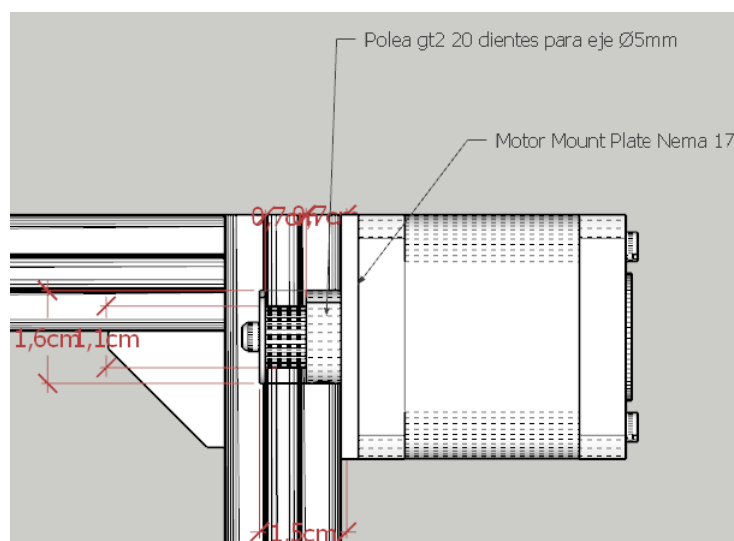


Figura 3.11: Segunda vista acotada desde arriba del eje X.

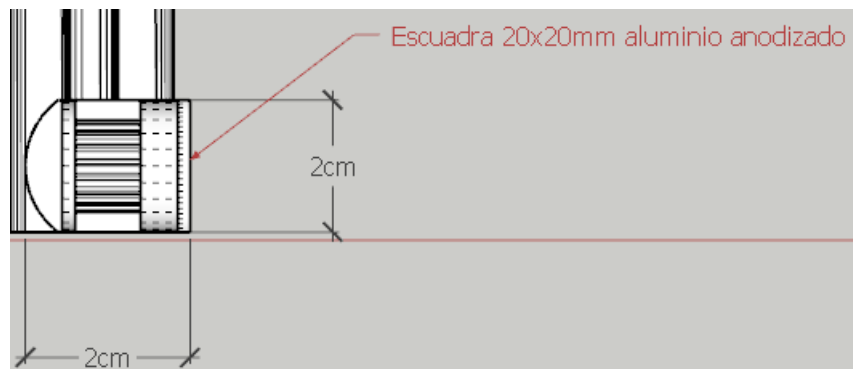


Figura 3.12: Tercera vista acotada desde arriba del eje X.

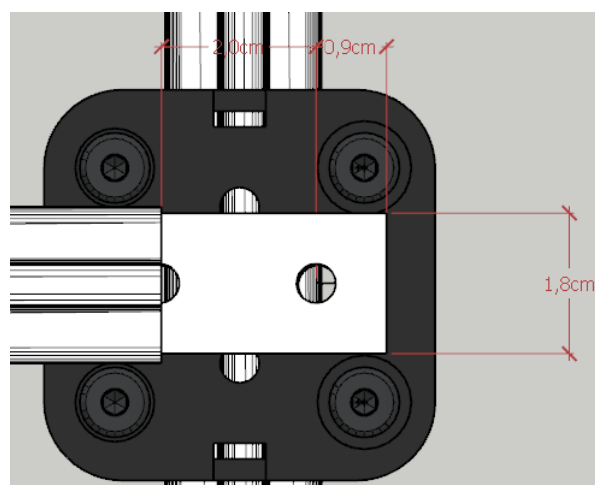


Figura 3.13: Cuarta vista acotada desde arriba del eje X.

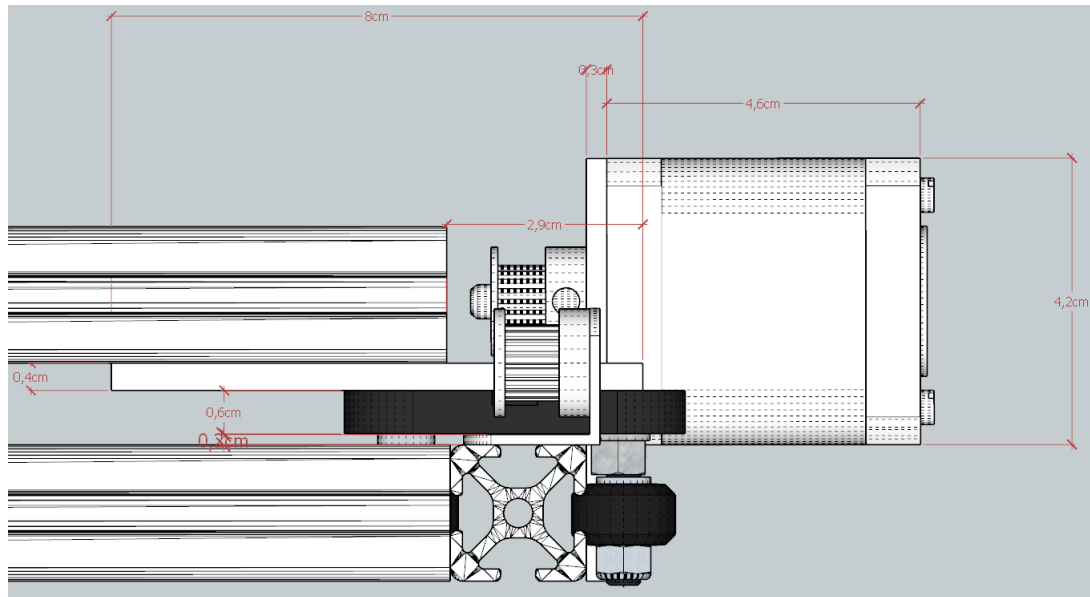


Figura 3.14: Vista acotada frontal del eje X.

3.2.2.3. Vistas acotadas del eje Y

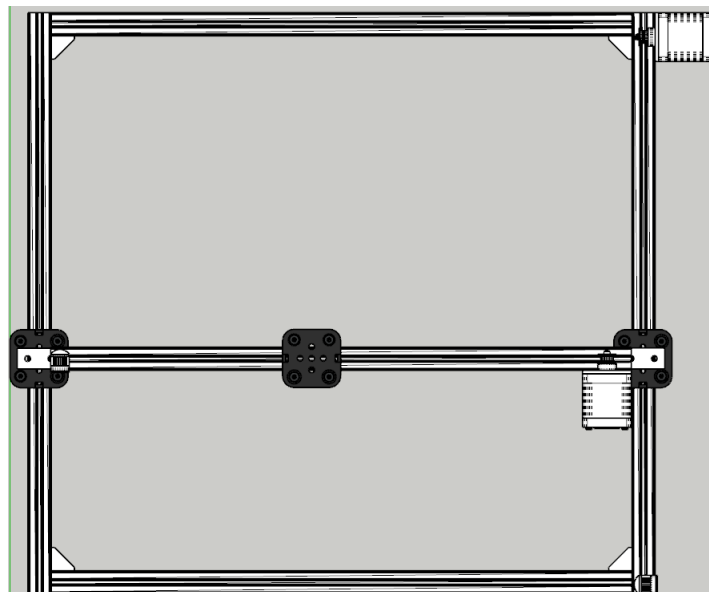


Figura 3.15: Vista desde arriba del eje Y

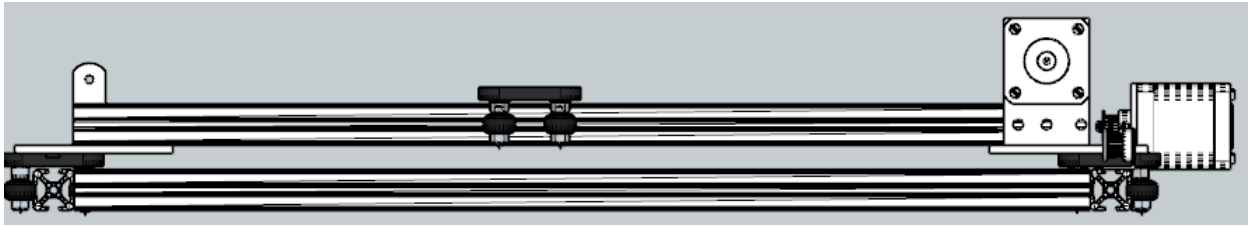


Figura 3.16: Vista frontal del eje Y

3.2.2.4. Vistas acotadas del eje Z

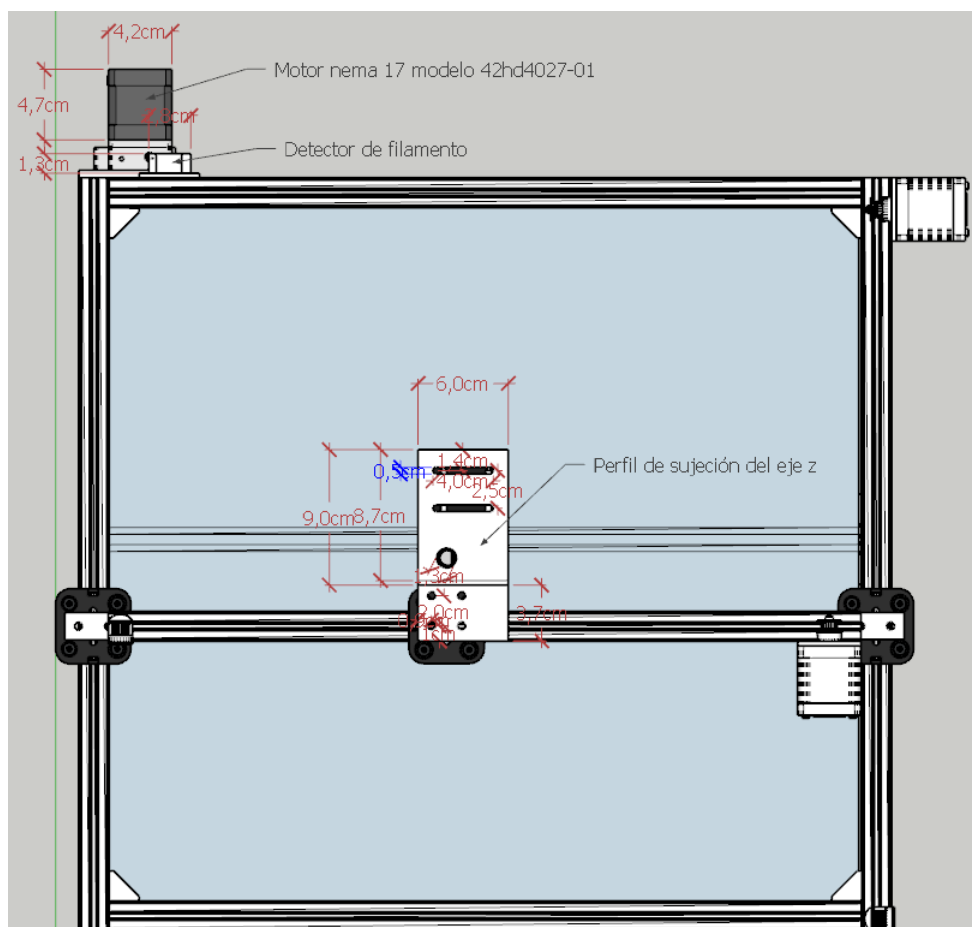


Figura 3.17: Vista acotada desde arriba del eje Z.

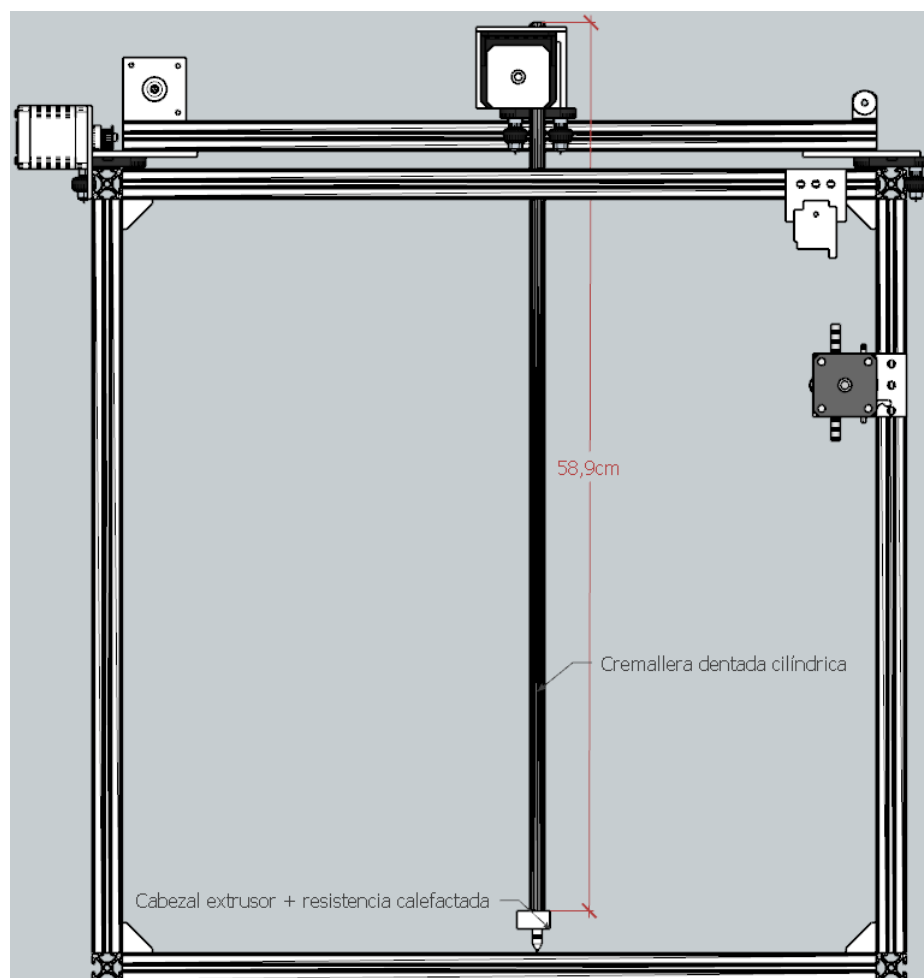


Figura 3.18: Vista acotada desde atrás del eje Z.

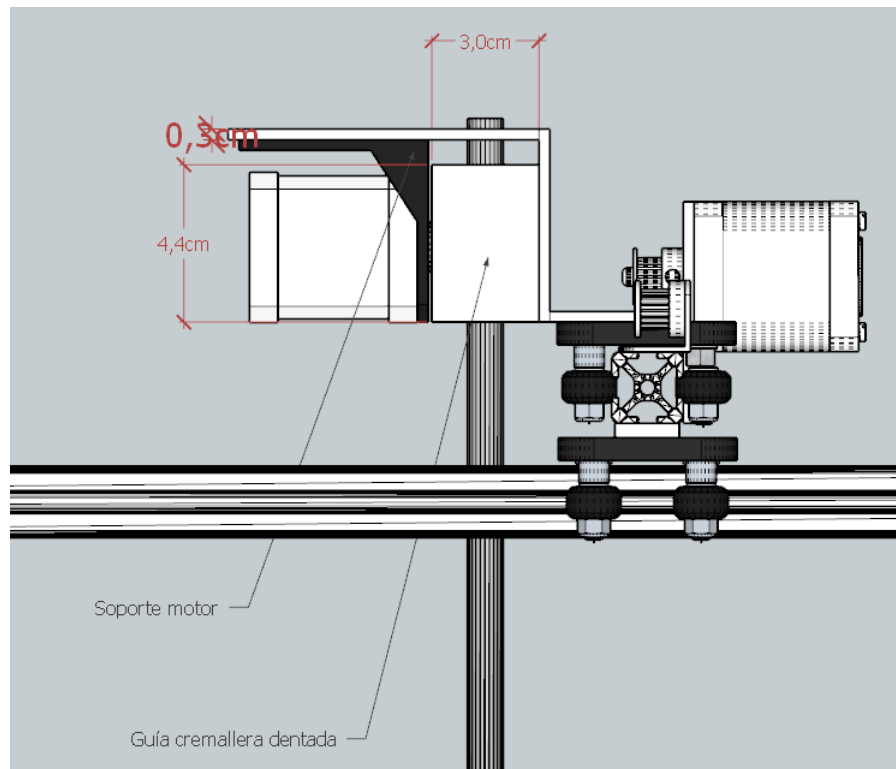


Figura 3.19: Vista acotada izquierda del eje Z.

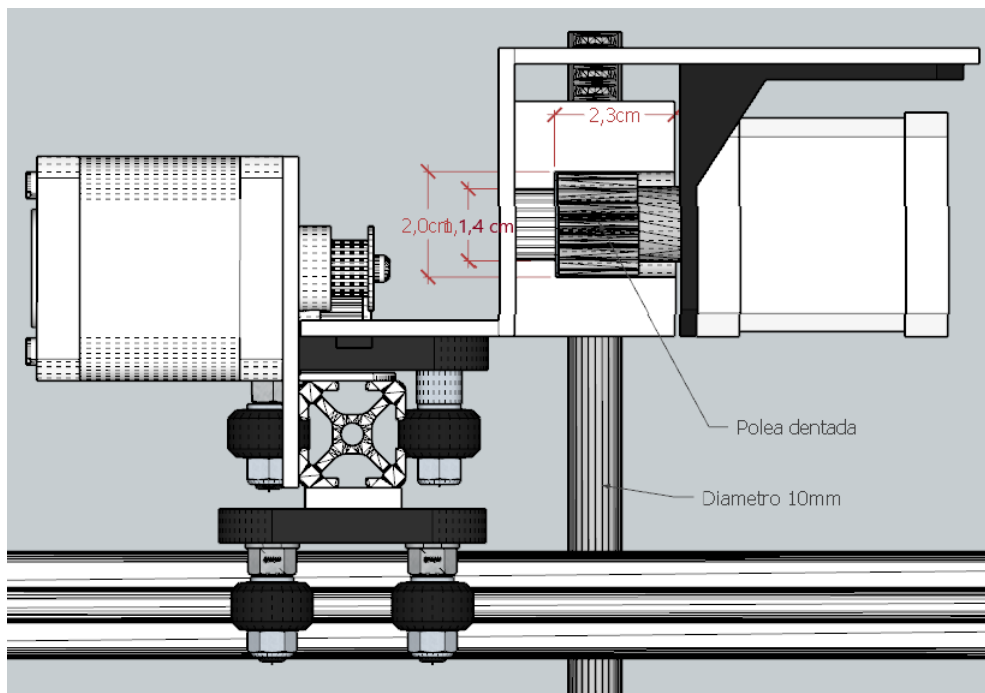


Figura 3.20: Vista acotada derecha del eje Z.

3.2.2.5. Materiales

Componente	Unidades
Perfil de aluminio T-slot 5 20x20x500	13
Tornillo allen cabeza ultra plana M5	75
Escuadra unión 20x20 aluminio para T-slot	28
Polea gt2 20 dientes para eje Ø5mm	4
Correa GT2 abierta	2
Escuadra 20x20 aluminio anodizado	2
Guía lineal/Patín	3
Motor Mount Plate Nema 17	3
T-nut con bola resorte	75
Tubo de teflón para filamento 1.75mm	1
Placa de unión 20x80 aluminio anodizado	1
Perfil de sujeción del eje z	1
Guía cremallera dentada	1
Cremallera dentada cilíndrica	1
Polea dentada	1
Tornillería y tuercas varias	18
Plataforma de impresión	1

Cuadro 3.1: Lista de materiales de la estructura.

En la sección 6 volverán a aparecer estos materiales junto con los enlaces de los distribuidores de éstos. En dichos enlaces viene contenido una descripción de los materiales más detallada y unas vistas acotadas.

Capítulo 4

Fase dos: Diseño hardware

4.1. Componentes

De los componentes mencionados en la tabla 4.1 lo siguientes son los que desempeñan un papel relevante en el funcionamiento de la placa de control:

- **RaspberryPi** Controladora de la impresora.
- **DRV8825** Variador de control para los motores.
- **Nema 17** Motores de la impresora. Se usan dos modelos distintos (42byghm809 y el 42hd4027-01)
- **MCP3002** Convertidor de analógico a digital.
- **CMX100D10** Relé de control para el extrusor.
- **Termistor NTC 100K** Componente de medición de temperatura.
- **Ventiladores** Ventiladores para regular la temperatura de los variadores y la controladora.
- **Botones** Botones para activar las interrupciones manuales.
- **Detector de filamento** Interruptor que detecta si hay filamento en la bobina.

Componente	Unidades
DRV8825	4
Detector de filamento	1
Funda Raspberrypi con ventilador integrado	1
Extrusor mk8(motor nema 17 modelo 42hd4027-01 + extrusor + termistor + resistencia)	1
Relé CMX60D5	1
Fuente de alimentación 12V 350W	1
Nema 17 modelo 42byghm809	3
Ventilador vapo 40x40x10 5V	2
Raspberry Pi 3 +B	1
Placa PCB de control	1
Resistencia 100K 2W	1
Zocalo 8 pines	1
Tornillería varia	36
Diodo 10A	1
MCP3002	1
Condensador electrolítico 25V	4
Cable 4 hilos	3
Cable 2 hilos	1
Conectores dupont + Pin macho dupont	40
Conectores termoretractiles	22
Varias regletas pin 2,54mm macho	40
Varias regletas pin 2,54mm hembra	40

Cuadro 4.1: Lista de componentes que comprenden la placa.

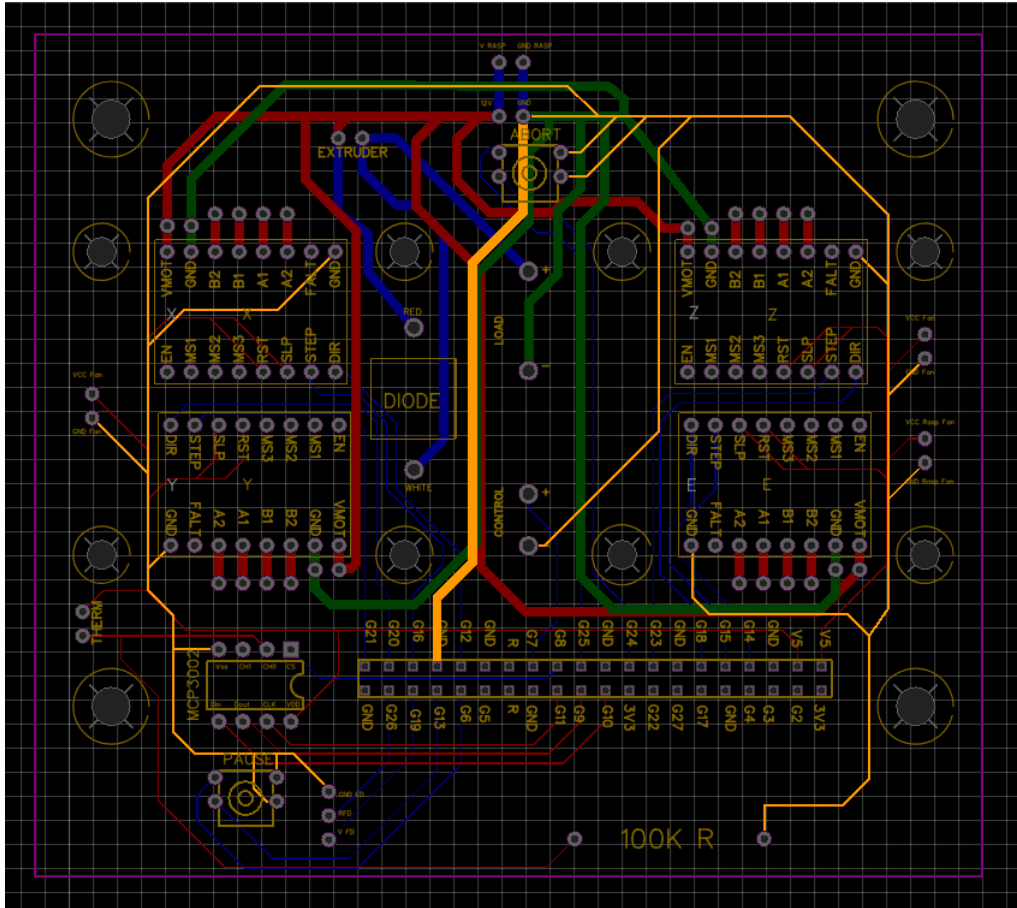


Figura 4.1: Esquema completo de la placa de control.

4.2. Implementación

4.2.1. RaspberryPi

La controladora usada para manejar la impresora es una Raspberry Pi [38], un pequeño ordenador de tamaño reducido que permite gran libertad para implementar hardware adicional.

Para interconectar la controladora con el resto de hardware, se implementaron una serie de pines dentro de la placa. En la figura 4.2 se puede ver donde están localizados estos pines en la placa.

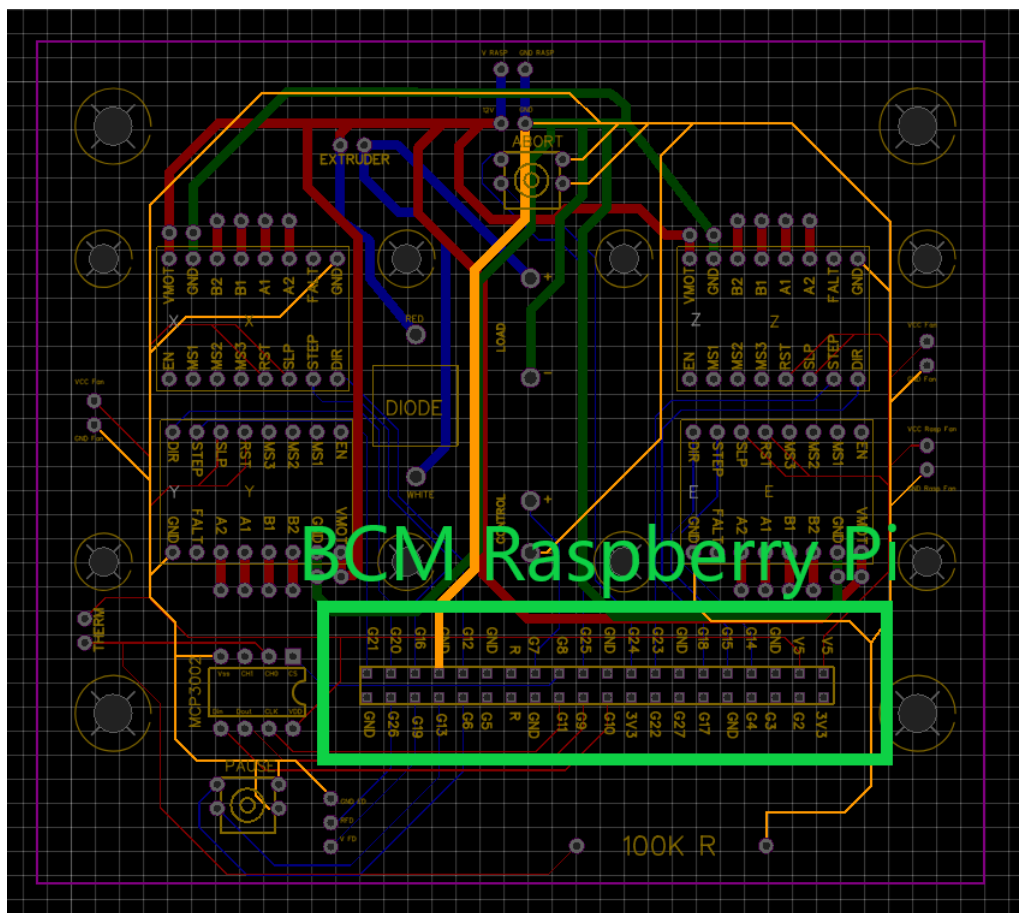


Figura 4.2: BCM Raspberry en la placa.

Desde los pines de la BCM (Broadcom stands for system on a chip channel) es desde donde se controlan todos los elementos que componen la placa. El término BCM hace referencia a la numeración que se utiliza para hacer referencia a los pines. BCM es la numeración que usa el chip de control de los pines. Existe otra numeración llamada GPIO. Ambas numeraciones hacen referencia a los mismos pines, pero con distintos números para cada uno de ellos.

En este pequeño ordenador corre un sistema operativo. Para la implementación de la impresora se usó el sistema operativo Raspbian, una versión basada en la distribución Debian adaptada a la controladora.

4.2.2. Nema 17

El motor paso a paso o motor bipolar es un dispositivo electromecánico que convierte una serie de impulsos eléctricos en desplazamiento físico del propio motor, lo que significa que

es capaz de girar una cantidad de grados, medidos en pasos, dependiendo de sus entradas de control. El motor paso es gobernado por impulsos procedentes de sistemas digitales. En nuestro caso por la Raspberry Pi [39].



Figura 4.3: Motor bipolar Nema 17.

Como se ve en la figura 4.4 un motor bipolar tiene varias bobinas enfrentadas entre si, de modo que cuando excitamos un par de ellas genera un campo magnético, una de las bobinas actúa como polo norte y la otra como polo sur, lo cual hace que el rotor magnético sienta este campo y se oriente en función de estas bobinas. Si vamos excitando los pares de bobina en un sentido u otro conseguimos que el motor avance en un sentido u otro. Este avance se consigue gracias a los variadores, los cuales implementan un circuito especial denominado puente en H doble para ir excitando las bobinas de manera ordenada y pareada y conseguir que el motor gire en le sentido y los grados que deseemos. En la sección 4.2.3 se explica con más detalle.

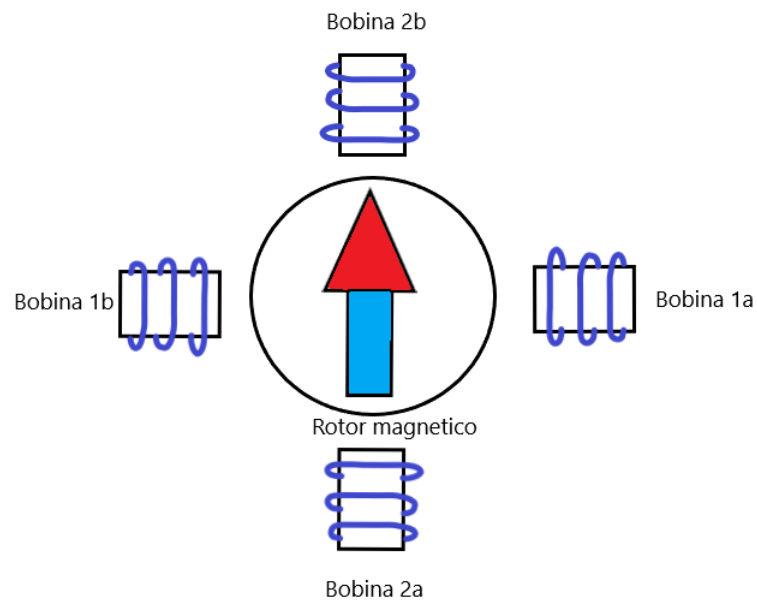


Figura 4.4: Esquema simplificado de las bobinas de un motor bipolar.

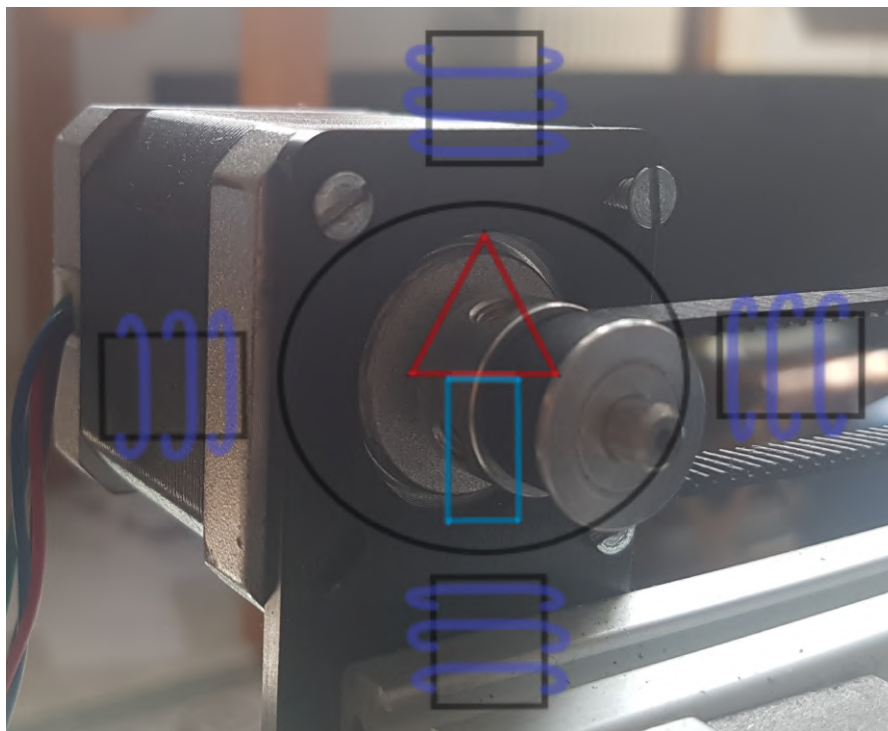


Figura 4.5: Motor nema 17 con sus bobinas y rotor.

En la impresora tenemos dos modelos distintos de motor bipolar Nema 17. La diferencia entre el modelo 42byghm809 y el modelo 42hd4027-01 radica en el paso que tiene el motor [40] [41].

Podemos entender el paso de un motor como el avance en grados que hace el motor, es decir, cuantos grados gira el rotor entre dos pares de bobinas colindantes. El primer modelo tiene un paso de $0,9^\circ$ y el segundo de $1,8^\circ$.

Esto implica que el primer modelo tiene una precisión mayor. Es por ello que este modelo se usa en los ejes X, Y y Z. Sin embargo, el motor que empuja el filamento no requiere de tanta precisión y por eso usamos otro modelo distinto.

Hay que recalcar que no solo difieren estos dos modelos en la precisión de pasos. También en sus características de voltaje y amperaje necesarios para operar. A mayor precisión de paso, mayor potencia necesitará el motor.

4.2.3. DRV8825

Un variador es una interfaz por la cual una placa controladora puede controlar un dispositivo externo a su arquitectura. El DRV8825 [42] es un variador para controlar un motor y se puede usar tanto para controlar dos motores de continua o un motor bipolar. A través de estos la Raspberry Pi puede ordenar al motor en que sentido girar, a que velocidad girar y cuantos pasos debe dar.

El variador tiene implementado en su interior un circuito denominado doble puente en H, el cual permite ir excitando los pares de bobinas en función de como queramos hacer girar el motor. El puente en H se llama así por la forma similar a una H que tiene el circuito. La utilidad de éste es la de invertir la polaridad del motor.

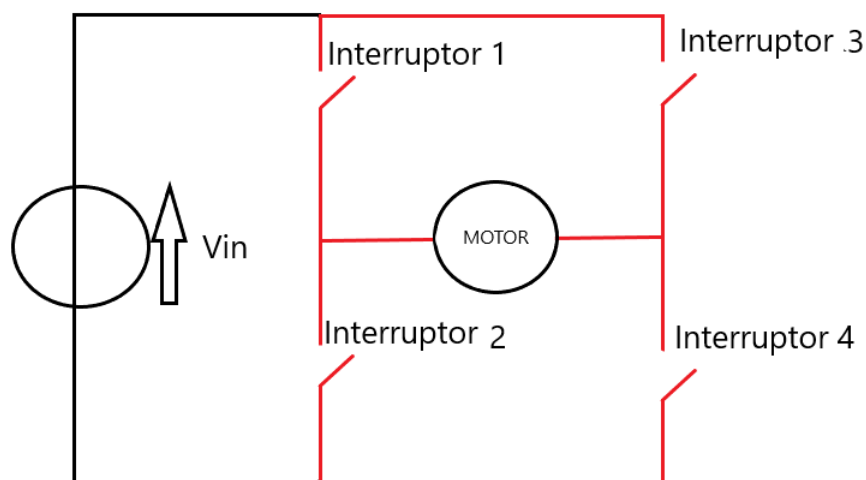


Figura 4.6: Simplificación de un puente en H.

El doble puente en H es la unión de dos circuitos en H. Cada uno de los puentes controla un par de las bobinas pareadas, como se puede ver en la figura 4.7. Si a la par nos fijamos en la figura 4.4 se ve la correlación de puente-bobina que tiene el motor y el variador.

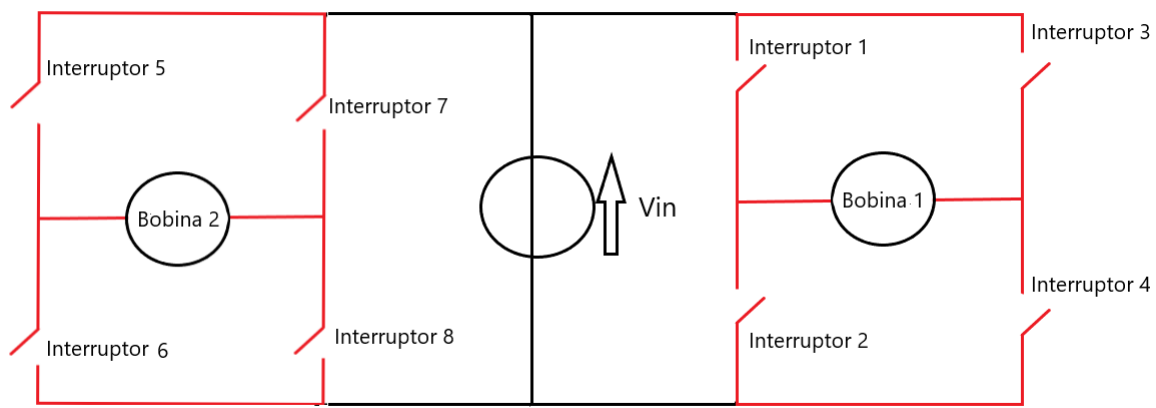


Figura 4.7: Simplificación de un doble puente en H.

Para controlar el sentido de giro del motor debemos ir abriendo o cerrando cada uno de los interruptores del circuito. Si nos fijamos en el circuito 4.7 vemos como hay ciertas combinaciones que producirían un corto circuito es por ello que para controlar el avance del giro solo es necesario controlar cuatro de los interruptores, ya que sus homólogos en el otro puente en H deben tener el mismo estado para no producir cortos. Por ejemplo, si el interruptor uno esta abierto, el interruptor cinco también debe estar abierto, si el cuatro esta abierto el ocho también y así sucesivamente. En la figura 4.2 se muestran estas combinaciones.

I1	I2	I3	I4	ESTADO
1	0	0	1	El motor gira en un sentido
0	1	1	0	El motor gira en el otro sentido.
0	0	0	0	El motor se detiene(sin oponer resistencia al movimiento)
1	0	1	0	El motor se detiene(oponiendo resistencia al movimiento)
0	1	0	1	El motor se detiene(oponiendo resistencia al movimiento)
1	1	0	0	Corto circuito.
0	0	1	1	Corto circuito.
1	1	1	1	Corto circuito.

Cuadro 4.2: Estados de un motor paso a paso.

Como se puede observar en la tabla 4.2 existen combinaciones dentro de un mismo puente que producirían un corto circuito, de modo que los interruptores siempre deben ir pareados 1-4 y 2-3.

Referido a motores bipolares, se considera que un paso es el mínimo avance rotatorio que puede hacer un motor. En el caso de uno de los motores NEMA 17 modelo 42byghm809 que se usan en la impresora, tienen un paso de 0,9 grados. Eso quiere decir que su avance mínimo por paso es de 0,9 grados que son los grados que separan dos pares de bobinas colindantes. Pero con este variador podemos controlar cuanto de paso queremos dar, es decir, que podemos hacer que el motor avance 1/16 de paso o incluso menos. Más adelante se explicará esto con detalle.

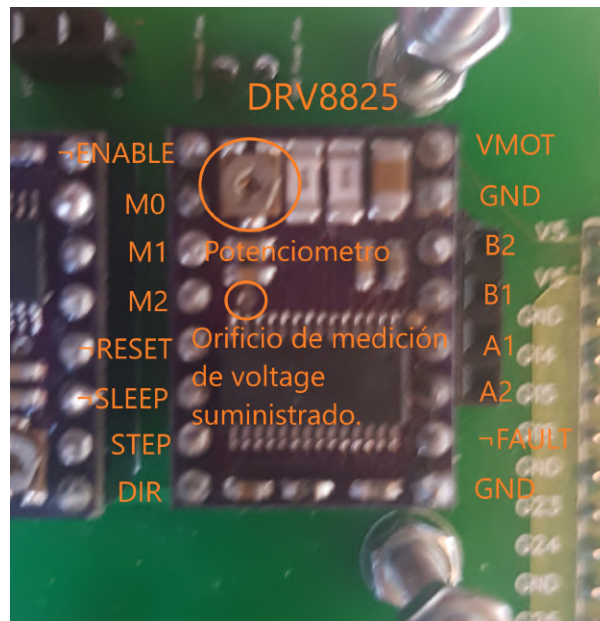


Figura 4.8: Esquema del DRV8825.

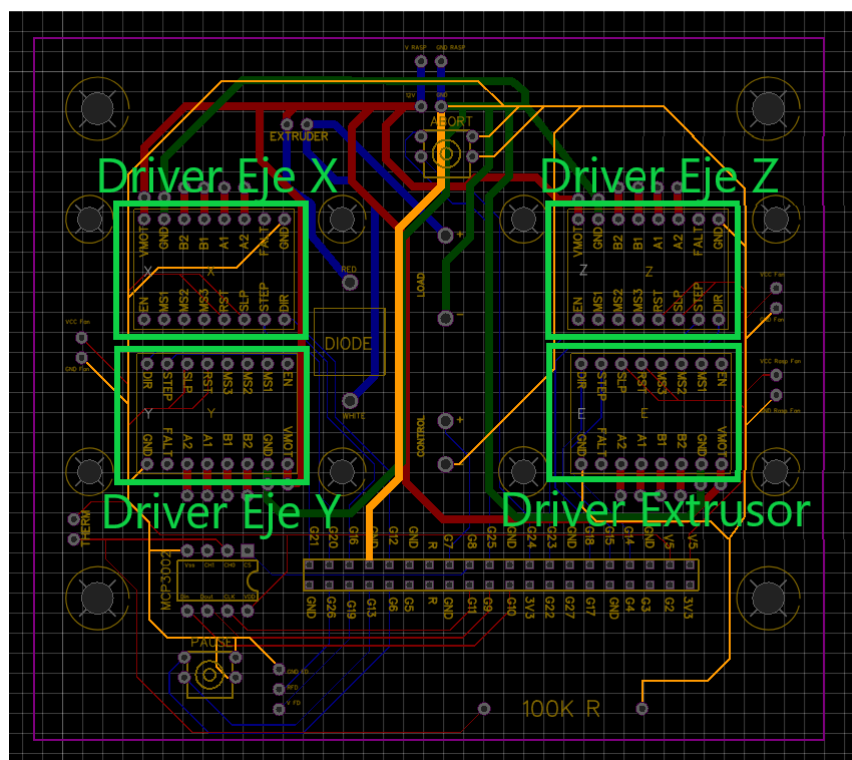


Figura 4.9: Ubicación de los variadores en la placa.

En la hoja de datos de este variador viene explicado con gran detalle cada una de las funciones de los pines y toda la información relativa a éste. No obstante, a continuación se explicarán

ciertas cosas sobre la configuración del variador que no vienen explicadas explícitamente en la hoja de datos o que no son tan evidentes.

4.2.3.1. Alimentación y avance rotacional por paso

Estos variadores permiten regular cuanta potencia recibirá por fase el motor que controla. Eso quiere decir que cada vez que se ordene al variador que haga avanzar un paso al motor, éste suministrará una cantidad de potencia. Ésta la podemos regular a través del potenciómetro que viene incluido.

Para saber cuanta potencia hay que suministrar al motor primero debemos ver la hoja de datos del motor que controlará. Por ejemplo, supongamos que tenemos un motor que requiere de 1,7 amperios de intensidad para operar. La fórmula que facilita el fabricante del variador para el cálculo es $A = V_{REF} * 2$. Siendo A la intensidad que necesita el motor y V_{REF} el voltaje de referencia que debe circular por el variador al suministrar energía al motor.

De este modo, obtendríamos un $V_{REF} = 0,85$ Voltios. Pero ahora nos queda decidir cuanto de paso queremos que de por fase. En la hoja de datos del fabricante viene el porcentaje de ese V_{REF} que tenemos que suministrar si queremos dar un paso completo, 1/32 de paso, etc. Supongamos que nosotros queremos dar el paso completo, es decir, no reducir el paso mínimo del motor. De modo que observando la tabla vemos el valor de 0,71 y aplicando el valor a la fórmula tenemos que $V_{REF} = 0,71 * 0,85$ y nos sale 0.6 voltios.

Eso quiere decir que cada vez que se ordene dar un paso al motor, si queremos que avance 0,9 grados debemos asegurarnos que el voltaje entre el orificio de medición de voltaje suministrado y cualquier tierra del variador debe ser 0.6 voltios.

También hay que añadir que si queremos hacer que el motor avance un 1/32 o algún valor distinto de una relación de 1 a 1 con el avance rotacional por paso del motor, debemos hacer uso de los pines M0, M1 y M2. En la hoja de datos viene estipulado cuales son los valores que deben tener estos pines en función de cuanto de paso queremos dar.

4.2.3.2. Control del motor

El variador tiene un control muy sencillo. El pin de dir indica en que dirección (Horaria o anti horaria) se desea girar el motor. Mientras que el pin step espera recibir una señal cuadrada. Se puede emular dicha señal cuadrada con la controladora, imponiendo nosotros el periodo de

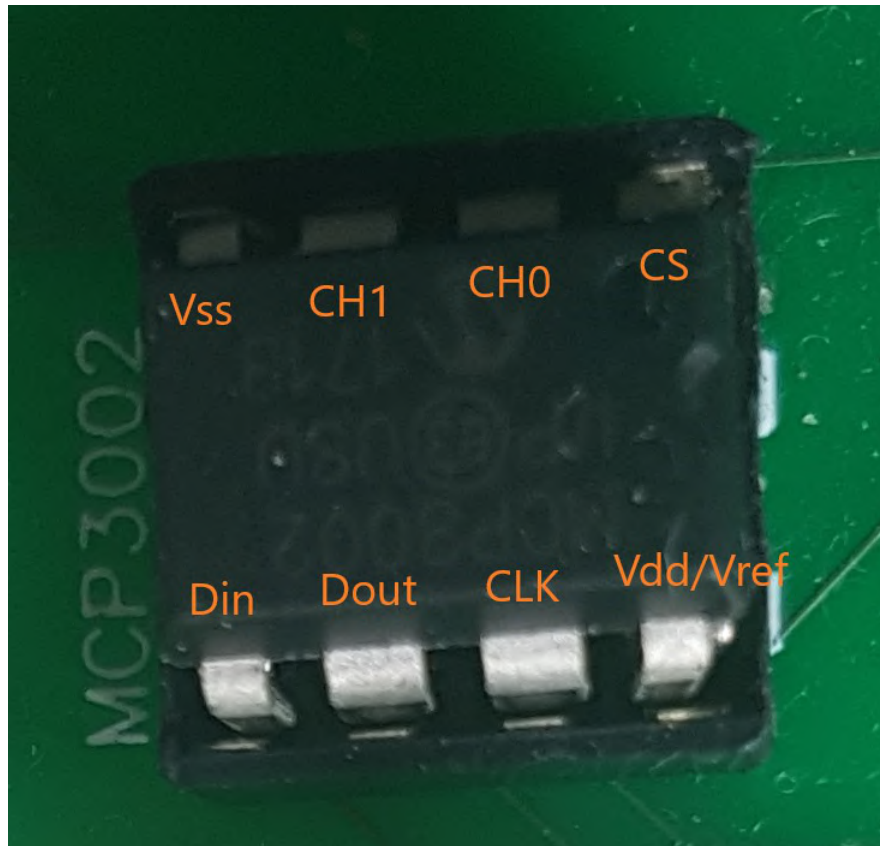


Figura 4.10: Esquema del MCP3002.

alto y bajo que tiene dicha señal. En función de cuál sea el periodo el motor ira más rápido o más lento. Esto quiere decir, que controlando el periodo de la señal se puede controlar la velocidad de giro del motor.

4.2.4. MCP3002

Este componente es un conversor de analógico a digital. Se entiende que una medida analógica es aquella que tiene valores continuos. Por ejemplo, la temperatura es un valor analógico ya que la temperatura puede tomar todos los valores pertenecientes al conjunto de los números reales. Una medida digital sería aquella que solo puede tomar valores discretos, por ejemplo, un bit. Un bit solo puede tomar valores de 0 o 1, sin valores intermedios.

Este componente permite hacer transformaciones de valores analógicos a digitales, para que una controladora que solo puede manejar valores digitales pueda tratar dichos valores analógicos. En el caso de la impresora, este componente sirve para hacer lecturas de temperaturas.

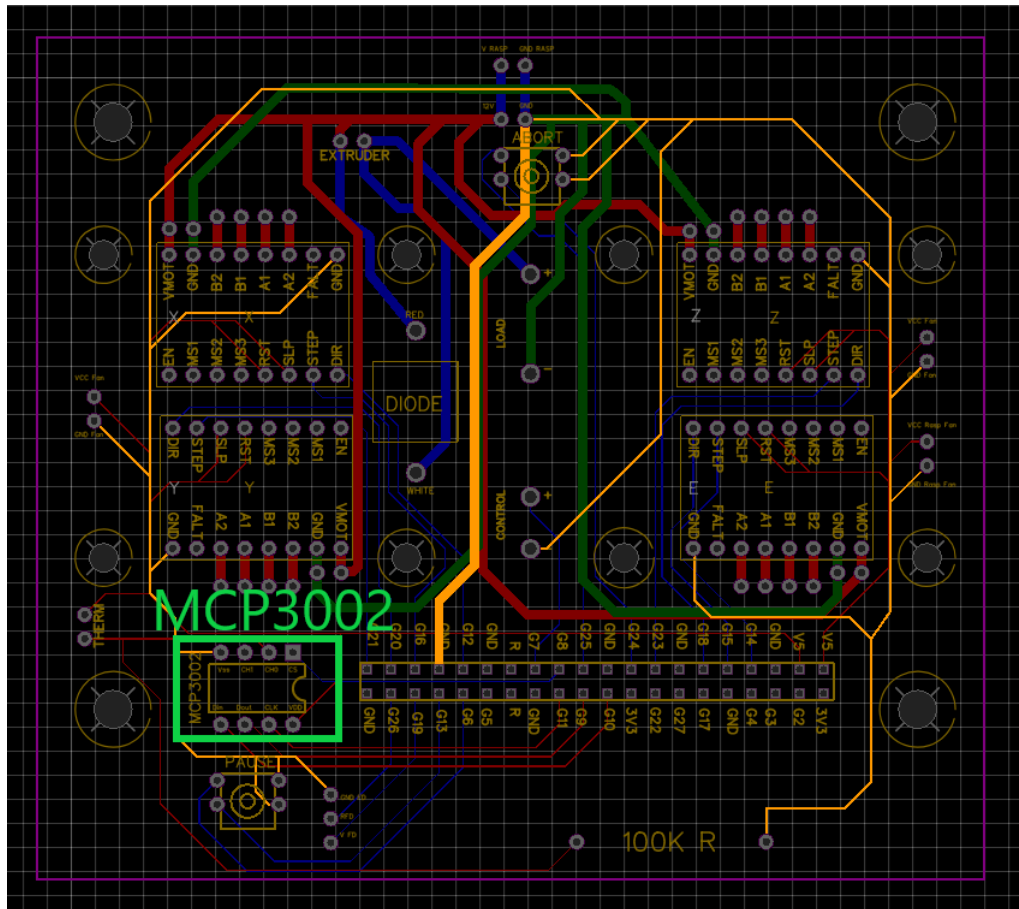


Figura 4.11: MCP3002 en la placa de control de la impresora.

A continuación se explica brevemente cada uno de los pines del componente que se muestran en la figura 4.10:

- **Vss** Tierra del componente.
- **CH1** Uno de los canales por el cual el convertidor recibirá la información analógica.
- **CH0** El otro canal.
- **CS** Pin de control de comunicación con la controladora. Selecciona y habilita a un esclavo.
- **Din** Pin de recepción de información enviada por la controladora.
- **Dout** Pin de emisión de información mandada a la controladora.

- **CLK** Reloj de la controladora para sincronizar la comunicación entre el convertidor y la controladora.

- **VDD/VREF** Este pin es a la vez la alimentación del componente y el voltaje de referencia para las lecturas analógicas. Más adelante se explicará esto con detalle.

Este componente se comunica con la controladora a través de el protocolo SPI (Serial peripheral interface). Éste es un protocolo de comunicación síncrono basado en maestro esclavo. Es decir, el controlador, el maestro, tiene una serie de periféricos, los esclavos, que esperan a las ordenes del controlador. Para que el maestro ordene algo a los esclavos, debe enviar una serie de bytes en los cuales se encuentran las órdenes que deben llevar a cabo los esclavos y los esclavos devuelven en esos mismos bytes la respuesta a la orden emitida.

En el caso del MCP3002 se envían dos bytes indicando que se quiere hacer. Dado que su función en la impresora es la de medir la temperatura, se le ordena que haga una lectura analógica de uno de los dos canales que posee el componente y este devuelve el valor leído en esos mismos bytes. De este modo obtenemos un valor digital que es la conversión analógica de la temperatura leída por un termistor conectado a uno de los canales. Posteriormente se aplicarán una serie de fórmulas para deducir a través de ese valor digital que temperatura ha leído [43]. La explicación de como se transforma este valor leído a temperatura esta en la sección 5.3.2.3.

El VREF es el voltaje de referencia de las lectura analógicas. Por ejemplo, supongamos que le damos 5 voltios de referencia. Eso quiere decir que el rango de devoluciones que puede hacer va desde 0 voltios hasta 5 voltios. Es el límite de voltaje que nos pueden devolver las lecturas de este componente.

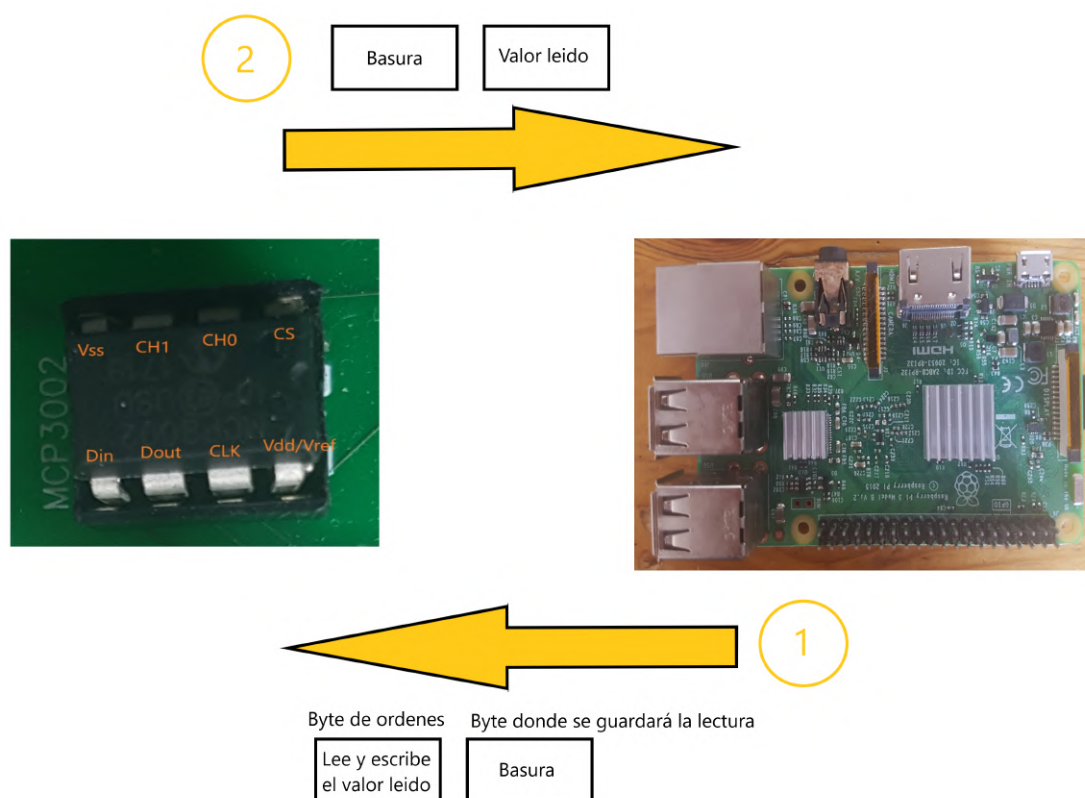


Figura 4.12: Esquema simplificado de la comunicación entre el esclavo MCP3002 y la controladora.

4.2.5. CMX100D10

Este componente es un relé de estado sólido. Un interruptor el cual no requiere de un mecanismo manual para dejar pasar o cortar la corriente, el cual es controlado digitalmente [44].

Basándonos en la figura 4.13, si dejamos pasar la corriente a través de los pines de control, un led lucirá y emitirá fotones. Esta emisión de fotones será percibida por el tiristor, que es un sensor de fotones, y permitirá el paso de corriente [45]. La ventaja de estos relés de estado sólido es que, al tener la conexión de control completamente separada del cableado de potencia, permite que por uno pueda fluir corriente continua y por el otro corriente alterna. Además, la vida útil de estos relés no es dependiente del número de veces que este se apague o se encienda, lo cual es más que ventajoso para el papel que desempeña que es la regulación de la temperatura. El relé, junto con el termistor, serán los encargados de controlar la resistencia calefactora.

se producen picos continuamente de modo que el diodo es necesario para el normal funcionamiento del relé.

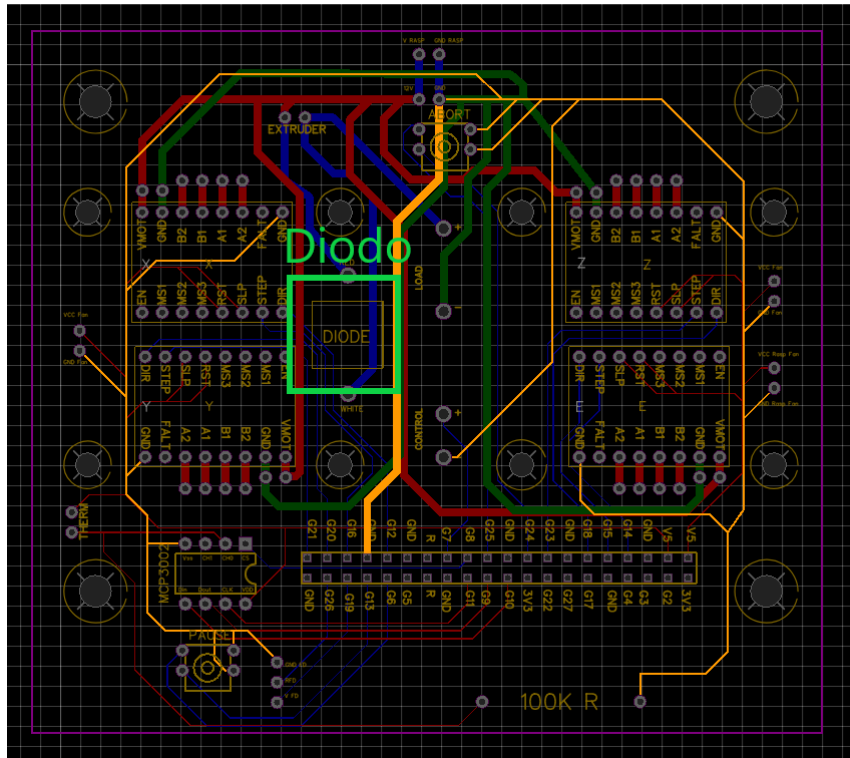


Figura 4.15: Diodo en la placa de control de la impresora.

4.2.6. Termistor NTC 100K

Un termistor es un sensor de temperatura. El funcionamiento de éste se basa en la variación de resistividad del material del que está hecho. La idea es sencilla, el material del que está hecho es un semiconductor, el cual, si recibe mayor temperatura ejercerá mayor resistencia al paso de la corriente, si por el contrario recibe menor temperatura, dejará pasar mayor corriente [46].

Con la tabla del fabricante que relaciona la resistencia ejercida por el termistor y la temperatura, podemos, obteniendo la resistencia, calcular cuál es la temperatura a la que está el termistor.

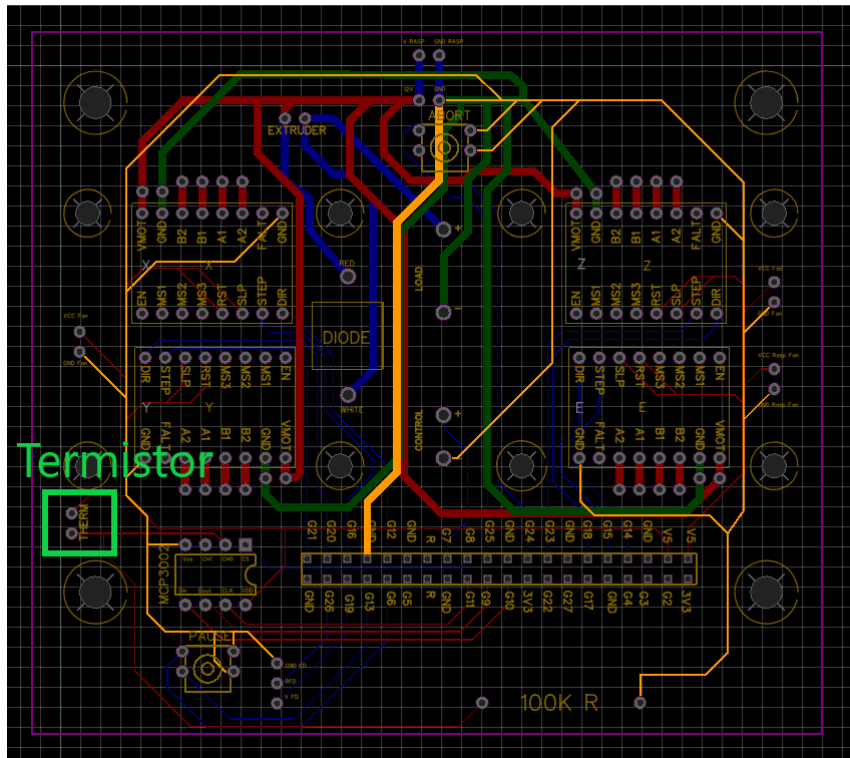


Figura 4.16: Conexiones del termistor en la placa de control de la impresora.

Para que el conversor de analógico a digital pueda realizar unas lecturas más limpias de los valores leídos por el termistor, se implementa un simple divisor de tensión con una resistencia igual a la del termistor. Generando una diferencia de potencial muy pequeña.

El termistor tendrá la función de medir la temperatura de la resistencia calefactora que se encuentra en el cabezal extrusor y la lógica de la impresora se encargará de que esté en un rango de temperaturas aceptables.

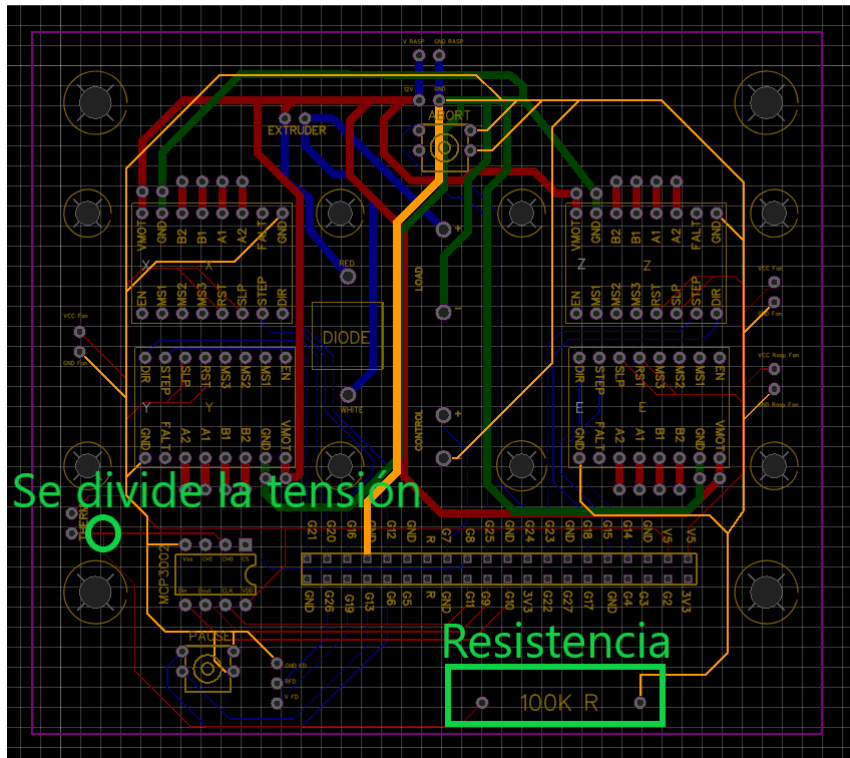


Figura 4.17: Divisor de tensión en la placa de control de la impresora.

4.2.7. Ventiladores

Dado que la impresora tiene unos tiempos de impresión muy altos, debemos proteger de temperaturas altas tanto la controladora como los variadores, los cuales reciben una intensidad más alta de la estándar. En la hoja de datos de los variadores se puede ver que se recomienda ventilación adicional si se supera la intensidad estándar.

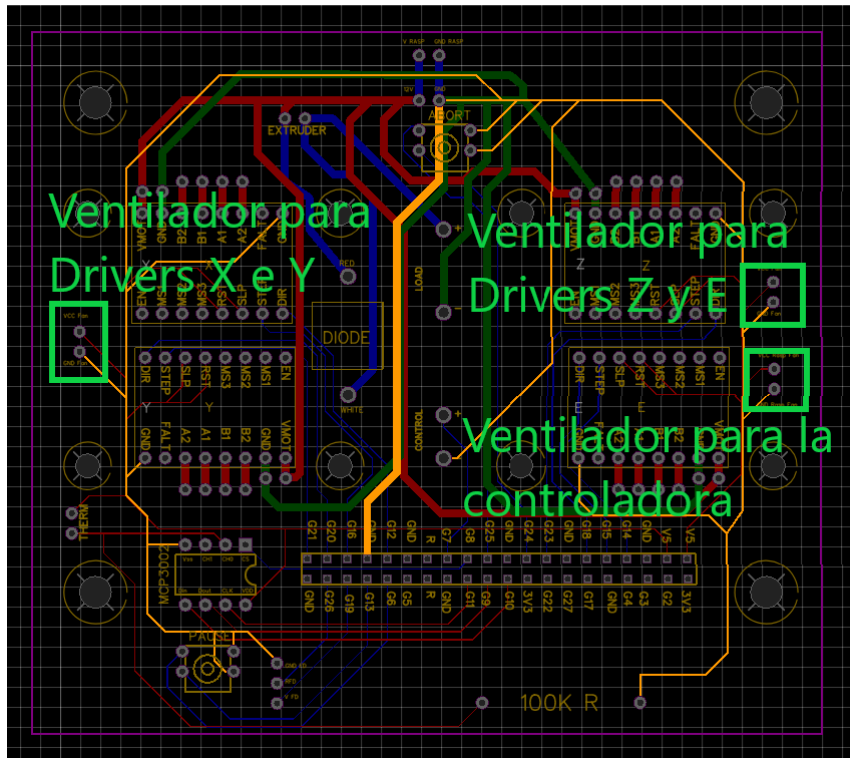


Figura 4.18: Ventiladores en la placa de control de la impresora.

4.2.8. Botones

Son unos interruptores manuales los cuales si se pulsan cortan la circulación de corriente, es decir, están implementados en alto.

Estos sirven para hacer una pausa o reinicio de la impresión y para abortar completamente la impresión actual. Si se pulsa una vez el botón de pausa, la impresión se pausará y si se vuelve a pulsar esta se reanudará.

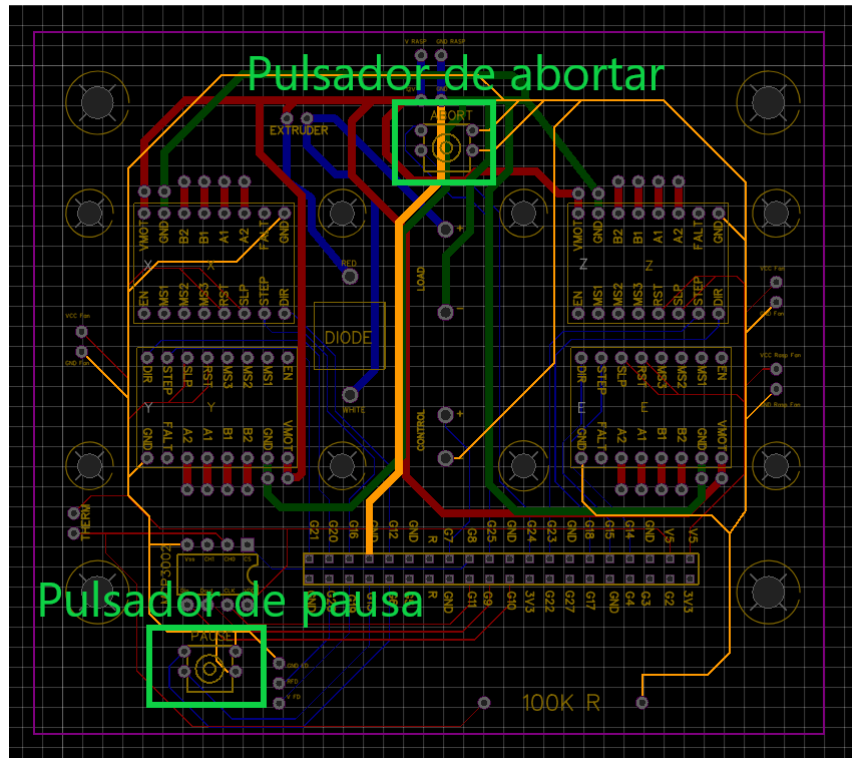


Figura 4.19: Pulsadores en la placa de control de la impresora.

4.2.9. Detector de filamento

Es un dispositivo que detecta si queda filamento en la impresora. Usa un interruptor mecánico para detectar si hay filamento o no.

Si el filamento pasa por medio del componente cierra el circuito y hace que el cable blanco, el de lectura, deje pasar la corriente. Si no contiene filamento, el interruptor separa la conexión de alimentación del componente y la de lectura haciendo que la lectura sea cero.

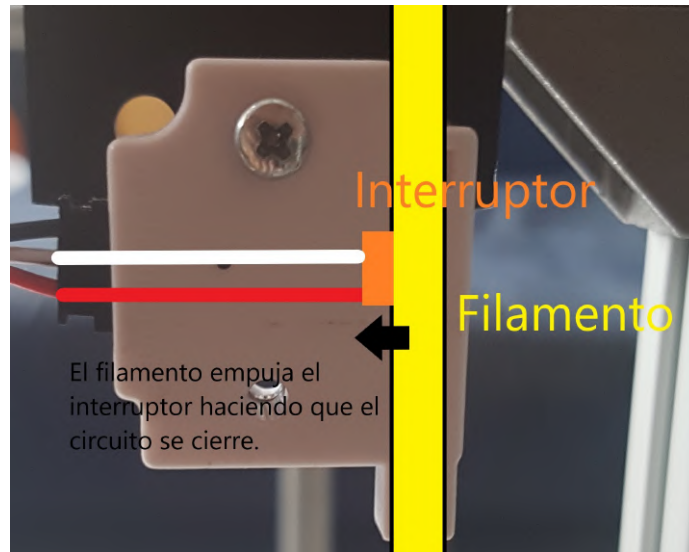


Figura 4.20: Detector de filamento activo.



Figura 4.21: Detector de filamento desactivado.

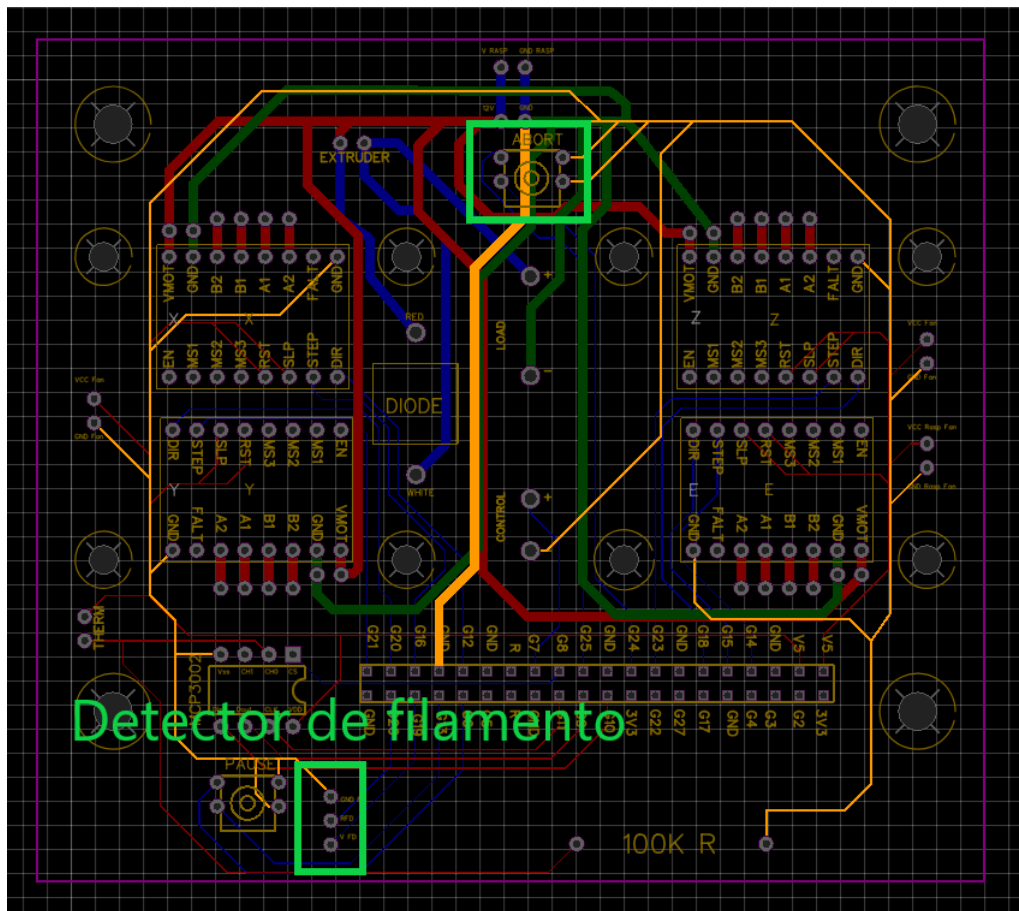


Figura 4.22: Conexiones del detector de filamento en la placa de control de la impresora.

Capítulo 5

Fase tres: Diseño software

Todo el software que se mostrará a continuación ha sido diseñado e implementado por mi.

5.1. Slicer

A grandes rasgos Slicer es un software escrito en C++ y es el encargado de calcular cuales son los puntos que componen la superficie de una figura dada.

El flujo de trabajo que sigue es el siguiente:

- **1.-** Con la herramienta OpenScad u otro entorno de desarrollo de diseño, generamos una figura tridimensional que estará representada por un archivo con extension .stl. Estos programas utilizan un algoritmo basado en la triangulación de Delaunay para representar el objeto modelado a través de triángulos. El archivo .stl resultante contiene la información de todos los triángulos que componen la figura. Una vez finalizada la modelación de la figura, esta será pasada como parámetro a Slicer.
- **2.-** Slicer recibe el archivo .stl (2.11) lo analiza y calcula cuales son las coordenadas de todos los puntos que componen la superficie de dicha figura. Éste devolverá dichas coordenadas ordenadas en función de la altura de los puntos.
- **3.-** Una vez Slicer haya finalizado su tarea, tomaremos los puntos calculados por éste para enviárselo a la impresora, la cual se encargara de ir pasando punto por punto para imprimir la figura.

Este programa debe su nombre a la técnica que utiliza para calcular las coordenadas de los puntos que componen la superficie de la figura. Slicer rebana el objeto por medio de planos paralelos al eje XY desde el plano con menor altura posible que corta a la figura hasta el mayor plano con altura posible que corta a la figura. El resultado de estos rebanamientos da una serie de rectas o trayectorias de las cuales calculará cuales son los puntos que las conforman y con ello obtendrá el resultado.

La separación entre estos planos de rebanamientos esta marcada por una variable denominada APRZ(Advance per rotation Z) que representa el mínimo avance lineal que hace la impresora en el eje Z. Este valor se calcula sabiendo cuantos grados gira el motor en un paso y el radio de la polea que hace desplazar el eje. De modo que si multiplicamos los ángulos, medidos en radianes, y el radio de la polea, obtenemos cuanto avance lineal se realiza en un paso de motor. Todos los ejes de la impresora poseen una variable APR, es decir, hay tres variables, APRX, APRY y APRZ.

El programa iterará la totalidad de los triángulos que componen la figura intersectandolos cada uno con los planos de rebanamiento paralelos al eje XY que intersecten con él. Es decir, no intersecta todos los planos de rebanamiento con todos los triángulos, solo intersectará aquellos que ya sabemos que van a hacerlo. Es fácil saber que planos paralelos al eje XY intersectarán con un triángulo dado, ya que si sabemos cuál es el punto más alto del triángulo y cuál es el más bajo, los planos con alturas intermedias entre estos dos valores, serán los que rebanen al triángulo.

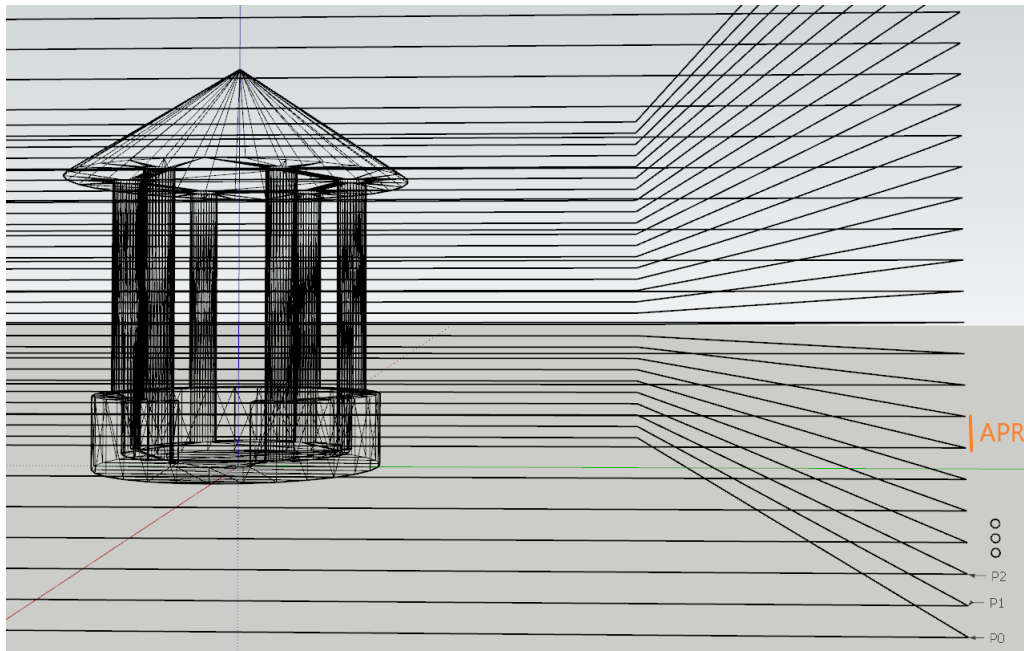


Figura 5.1: Planos de rebanamiento.

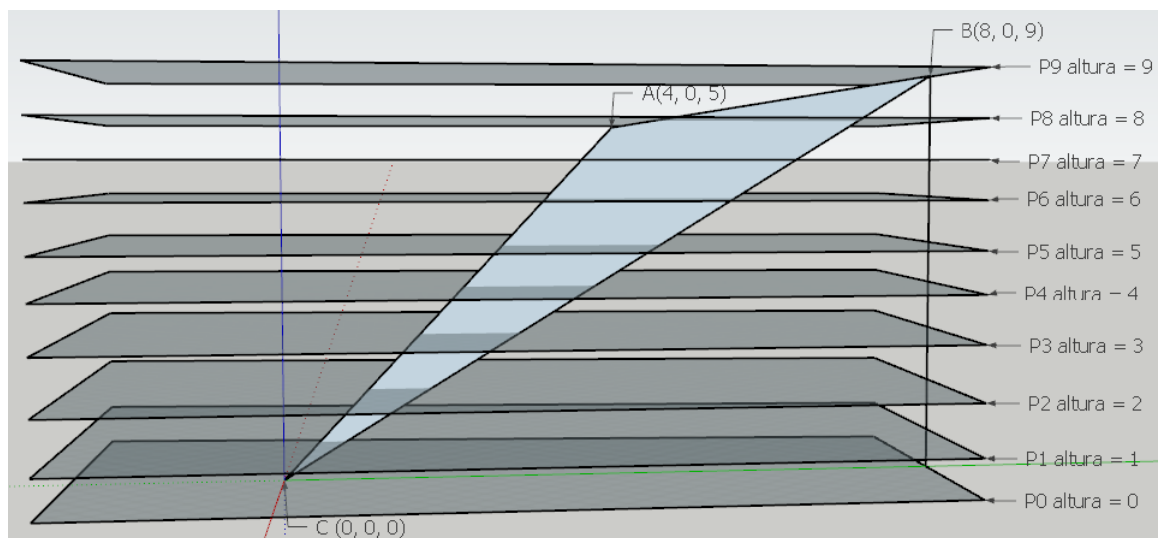


Figura 5.2: Cálculo de los planos que cortan al triángulo (APRZ = 1).

5.1.1. ¿Dónde opera?

Dentro del esquema general del sistema, Slicer operaría en la parte del cliente, es decir, en el PC del usuario. Este se ejecutaría a través de la interfaz visual Nao. Una vez el usuario ya haya establecido conexión y se haya cerciorado de la disponibilidad de la impresora, se lanza Slicer

para calcular las trayectorias/puntos que componen la figura y estos se enviarán a la impresora.

A continuación se expone el lugar que ocupa el Slicer dentro del esquema general:

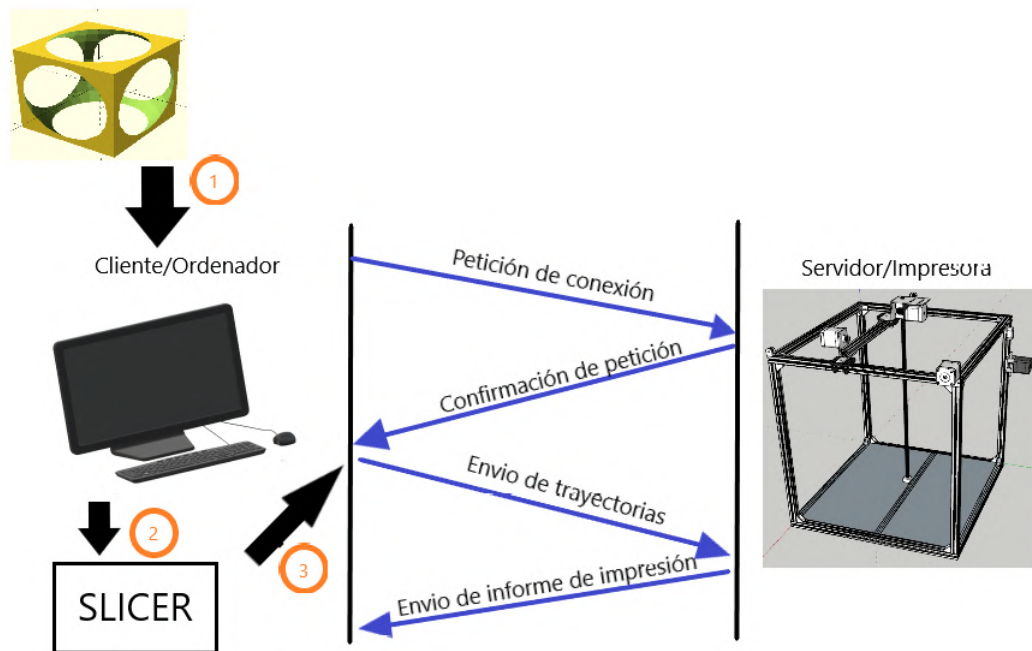


Figura 5.3: Lugar que ocupa Slicer dentro del esquema general de sistema.

5.1.2. Estructura y composición

En este apartado se explica cuál es la estructura y composición del Slicer. El Slicer está compuesto por un total de 6 paquetes: **BaseUniTest**, **Configuration**, **Figure**, **Geometry**, **Main** y **Math**.

A continuación se expondrán con más en detalle cada una de éstos.

5.1.2.1. BaseUniTest

La utilidad de este paquete es dotar a las demás clases del proyecto que lo requieran de la capacidad de analizar los resultados obtenidos después de realizar los cálculos pertinentes y

comprobar si estos son correctos.

Esta carpeta esta compuesta por dos clases:

5.1.2.1.1 Clase Test

Esta clase está compuesta por una serie de condiciones, las cuales son un tipo definido `pair[string, bool]`. El valor `string` hace referencia al par de valores que se están comparando y el booleano determina si la restricción sobre ese par de valores se están cumpliendo.

El `string` sirve para que, en caso de que no se cumpla la restricción muestre el par de valores que no la cumplen.

Cuando queremos comprobar varias cosas, cargamos en esta clase todas las condiciones que debe satisfacer y las insertamos dentro de la clase `BaseUniTest`.

5.1.2.1.2 Clase BaseUniTest

Esta clase está conformada por uno o más tests. Se encarga de revisar que todas las condiciones impuestas en los tests se cumplan y en caso de no ser así, muestra cuales de las condiciones han fallado y cuál es el ratio de éxito de los tests en total.

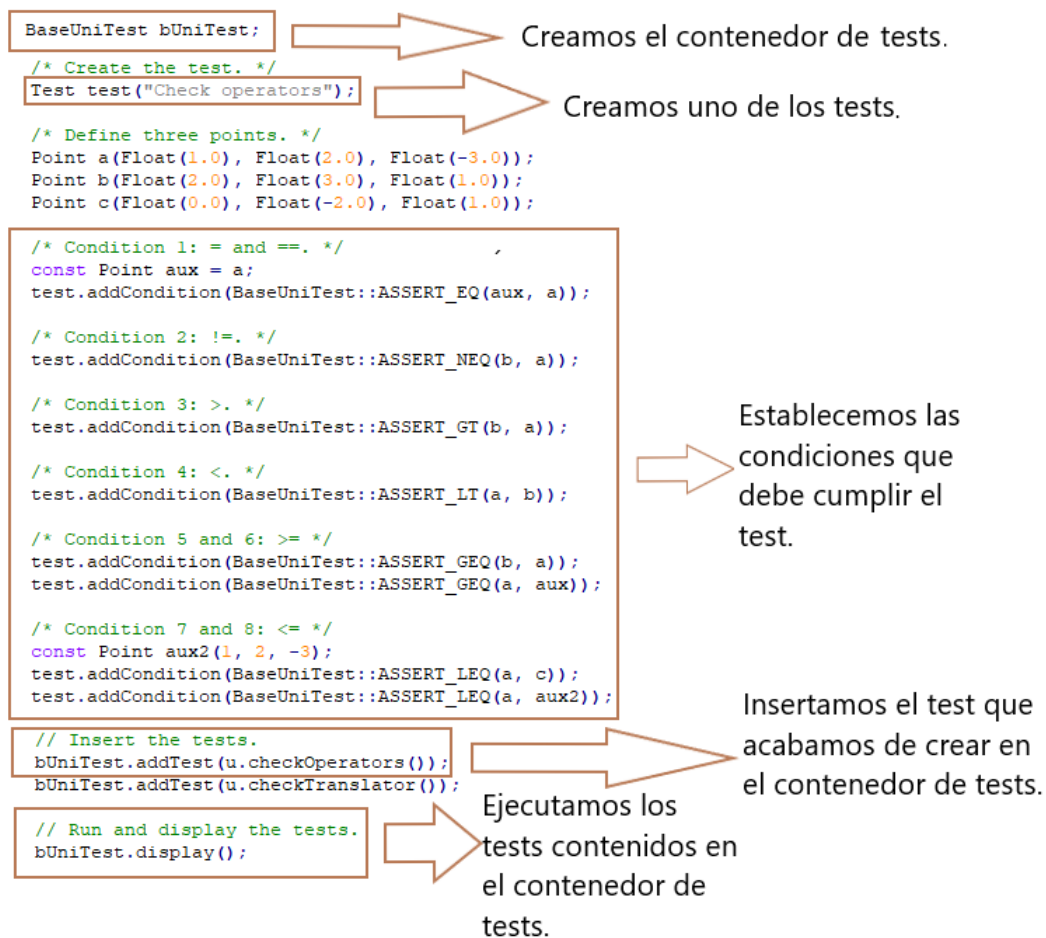


Figura 5.4: Ejemplo de Test.

```

<terminated> FigureMaker
#####
##    RUN TEST(S)    ##
#####
-----
TEST Check operators:
All conditions are OK!
-----
TEST Check translator:
All conditions are OK!
-----
Test Ratio: 100.00%

```

Figura 5.5: Ejemplo de ejecución satisfactoria.

```

<terminated> FigureMaker [C/C++ Applicati
#####
##    RUN TEST(S)    ##
#####
-----
TEST Check operators:
Not all conditions have been passed:
Condition 1: 1 2 != 2 3
Condition 2: 1 2 == 1 2
-----
TEST Check translator:
All conditions are OK!
-----
Test Ratio: 50.00%

```

Figura 5.6: Ejemplo de ejecución insatisfactoria.

Estas clases son de gran utilidad ya que si queremos corroborar que cualquier clase ha sido implementada correctamente solo tendremos que ejecutar estos tests imponiendo las restricciones que deben cumplir. Éstas también se podrían usar en cualquier proyecto ya que han sido implementadas a través de templates, de modo que se adapta al cien por cien a la implementación del proyecto en el que esté contenido, la única restricción es que si se comparan dos clases deben tener implementados operadores de comparación y siempre se deben comparar dos clases del mismo tipo.

Todas las clases que se mencionen a continuación tienen un paquete con su propia implementación de estos tests para comprobar que funcionan correctamente.

5.1.2.2. Configuration

Este paquete está compuesto por un archivo de cabecera y un archivo de código llamados Configuration.h y Configuration.cpp respectivamente.

Éstos contienen la información relativa a la impresora, es decir, los valores que determinan las propiedades de la impresora.

A continuación se expone cada una de las variables de configuración:

- **ADVANCE-PER-ROTATION** Cantidad de mm lineales que avanza un eje cada vez que un motor da un paso. Dado que cada eje puede tener un avance lineal distinto, cada uno de los ejes coordenados tiene su propia variable.
- **ADVANCE** Relación entre ADVANCE-PER-ROTATION y el diámetro de extrusión de la impresora. Esta medida será la que determine la distancia real que debe separar a todo

punto de sus colindantes. Dado que cuando extruimos el filamento de plástico este tiene un grosor, debemos tener en cuenta cuál es dicho grosor y dividiendo el diámetro de extrusión entre ADVANCE-PER-ROTATION obtenemos cuál es la distancia que debe recorrer en mm la impresora al imprimir el siguiente punto colindante.

- **INFINITE** Estas son unas variables que determinan cuál es el infinito de la impresora. Supongamos que una impresora tiene unas medidas de 30x20x40cm de impresión, eso quiere decir que cualquier medida fuera del rango se considera infinita, serían valores no válidos. De modo que para calcular estas variables lo único que debemos hacer es coger el valor máximo de las medidas de la impresora y sumarle uno. De ese modo, cualquier valor en el eje coordenado que sea igual o superior a ese infinito calculado se considerará inválido. En este caso, sería INFINITE-X = 31, INFINITE-Y = 21 y INFINITE-Z = 41.
- **PRINTER-SIZES** Contenedor donde guardamos las 3 medidas de los tres ejes(X, Y, Z) de la impresora.
- **EPSILON** Dado que trabajamos con valores en coma flotante, la imprecisión a la hora de comparar dos valores u operarlos se acentúa, de modo que tenemos que crear un valor epsilon de tal modo que si por ejemplo comparamos dos valores y la diferencia entre estos es menor que epsilon, consideraremos que dichos valores son iguales.
- **PRECISION-LIMIT** Éste determina cuanto de preciso queremos que sean las comparaciones entre dos valores. Cuanto mayor sea este número, mayor precisión tendremos y nuestro valor EPSILON será menor.
- **PATH** Ruta donde se encuentra el archivo .stl.

Además hay una función llamada loadConfiguration, ésta se encarga de cargar todos los valores mencionados anteriormente y asignárselos a las variables.

5.1.2.3. Figure

Este paquete está compuesto por la clase Figure.

La clase Figure contiene una lista de los triángulos que componen la figura y un mapa que contiene el conjunto de rectas/trayectorias que se generaron tras realizar la intersección entre los planos de slice con la figura.

El constructor de la clase recibe como parámetro todos los triángulos que componen la figura y los carga dentro de su lista de triángulos para que posteriormente, se llame al método slicer. Este recorrerá todos los triángulos de la lista intersectándolos con sus respectivos planos de slice.

5.1.2.4. Geometry

Este paquete contiene el core del programa, aquí es donde se calculan todas las intersecciones y donde están contenidos todos los elementos geométricos que jugarán un papel en el Slicer.

5.1.2.4.1 Equations

En este paquete hay dos clases. Ambas son representaciones de ecuaciones de planos en el espacio tridimensional. Una de ellas es la ecuación implícita y la otra es la ecuación paramétrica.

La clase que representa la ecuación implícita se llama `ImplicidEq` y tiene una lista de valores en coma flotante que guarda las cuatro componentes de la ecuación, cada elemento representa un valor en los ejes coordenados y el valor constante de la ecuación:

$$0(x) \ 1(y) \ 2(z) \ 3(d)$$

Además tiene varios constructores para inicializar la ecuación y un método `getter` para obtener cualquiera de los cuatro valores que componen la ecuación.

De manera análoga, la clase que representa la ecuación paramétrica se llama `ParametricEq` y esta compuesta por una lista (el atributo se llama `pEq`), cada uno de los elementos de ésta representa los valores en los ejes:

$$x = pEq[0][0] \ t + pEq[0][1] \ y = pEq[1][0] \ t + pEq[1][1] \ z = pEq[2][0] \ t + pEq[2][1]$$

Del mismo modo que la anterior clase, esta contiene varios tipos de constructores y un método `getter` para acceder a los parámetros que componen la ecuación.

5.1.2.4.2 Line

Este paquete está compuesto por la clase `Line`.

La clase `Line` representa las rectas en el espacio cartesiano. Está compuesta por un punto denominado `begin` el cual es el inicio de definición de la recta, un punto denominado `end` que es el fin de definición de la recta, un vector que representa la dirección de ésta, el conjunto de

puntos desde begin hasta end que conforman la recta, la ecuación paramétrica que define la recta y el tamaño de la recta.

La clase contiene métodos getter para todos los elementos que la componen, incluyendo un getter adicional que nos devuelve cuantos puntos contiene la recta.

Además, la recta contiene dos operadores, uno de ellos es el operador menor, el cual determina si una recta es menor que otra en función de su punto final. Si el punto final de una recta tiene una mayor altura y/o un mayor largo y/o un mayor ancho que el punto final de otra recta, se considera que la primera recta es mayor que la segunda. El motivo de este operador es para cuando ordenamos las rectas. Dado que cuando la impresora tenga que imprimir la figura, deberá saber qué orden debe seguir. Éste es el modo por el cual sabe en qué orden debe imprimir las rectas.

El segundo operador es el de igualdad. Este sirve para evitar que se repitan rectas o si dos rectas son prácticamente la misma, unificarlas y así evitar trabajo a la hora de realizar la impresión. El método revisa si las fórmulas paramétricas de ambas rectas son equivalentes, de ser así significa que son paralelas. Si además dichas rectas son secantes, significa que representan la misma recta de modo que se vuelcan los puntos pertenecientes de la segunda recta a la primera. También puede ocurrir que ambas rectas sean paralelas pero no secantes, en tal caso, si la distancia que las separa es menor que ADVANCE, también considerará que ambas rectas son las mismas.

Por último, la clase contiene un conjunto de métodos que sirven para calcular la intersección entre dos rectas. Estos métodos, resuelven el sistema de ecuaciones por matrices que se genera entre las dos rectas, y posteriormente se comprueba si la solución es válida y si está entre los límites de ambas rectas.

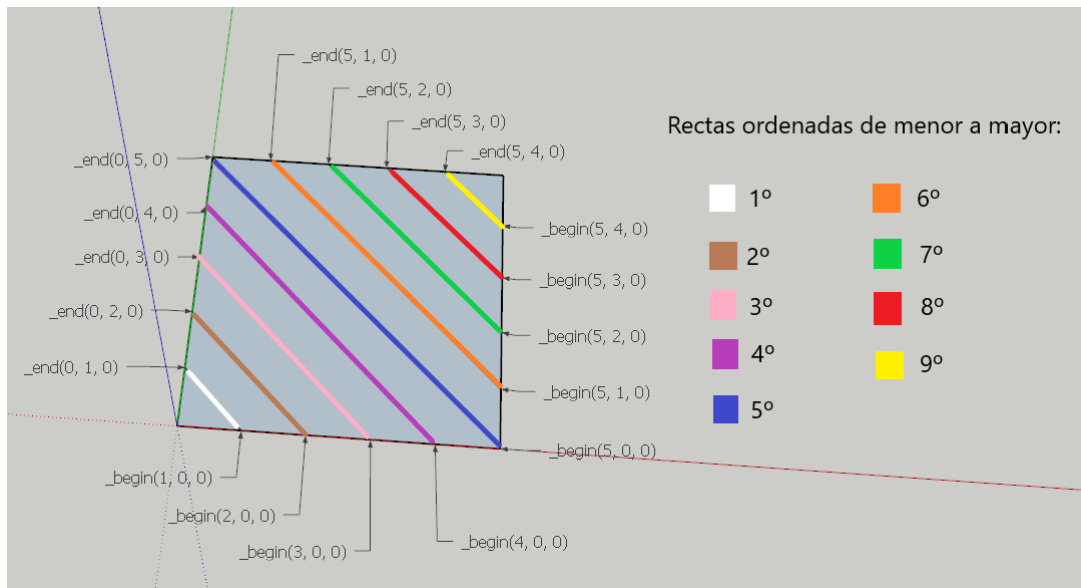


Figura 5.7: Ejemplo de ordenación de rectas.

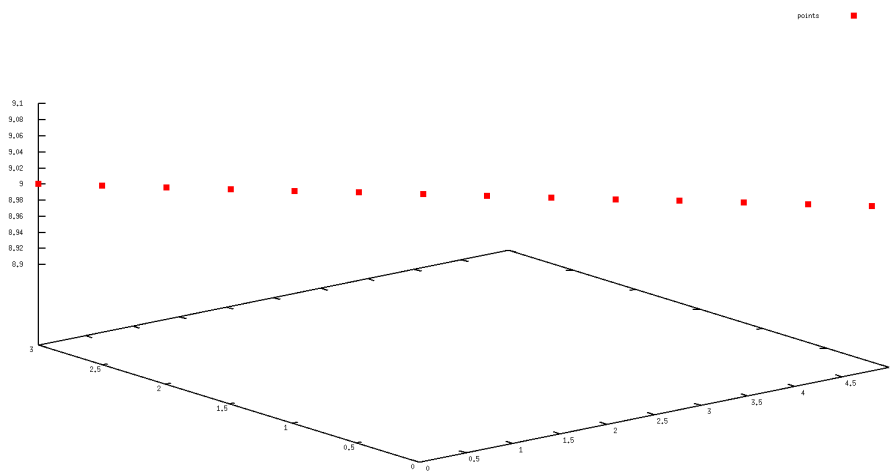


Figura 5.8: Ejemplo de puntos que componen una recta.

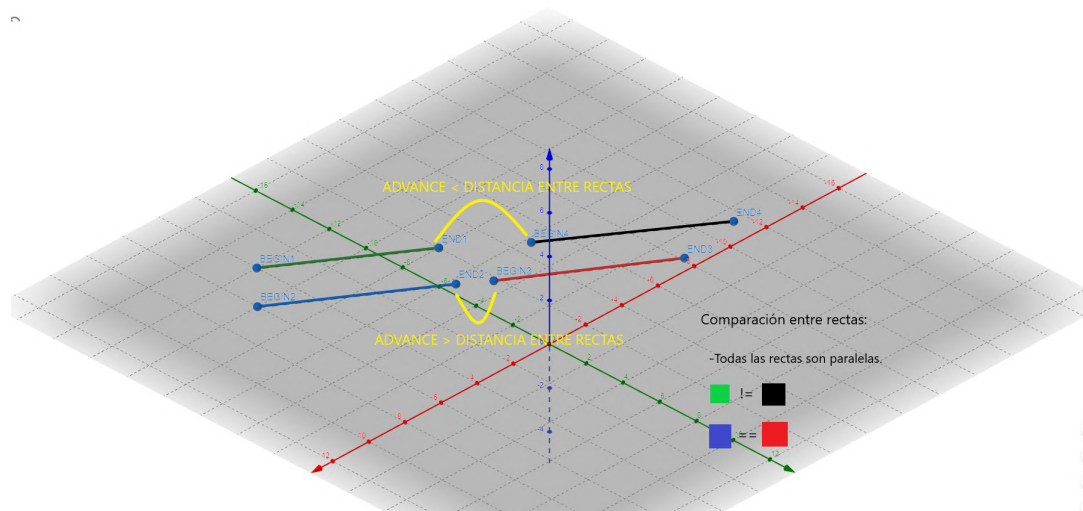


Figura 5.9: Ejemplo de comparación entre rectas.

5.1.2.4.3 Plane

Este paquete está compuesto por la clase Plane que representa los planos en espacio cartesiano. Esta compuesta por la ecuación implícita que lo representa. Además contiene una serie de constructores para construir planos a través de una ecuación implícita, de dos rectas, un punto y una recta y con un valor real. En el último caso se construiría un plano, con altura en el eje z igual al valor real pasado por parámetro, paralelo al eje XY, lo cual es bastante útil para construir los planos de slice mencionados anteriormente.

También tiene otros métodos para revisar la pertenencia de una recta al plano, un punto al plano y revisar si el plano es paralelo a los ejes XY, YZ o XZ.

Por último, la clase contiene un método para calcular la intersección entre dos planos, el cual resuelve el sistema de ecuaciones por medio del método de sustitución.

5.1.2.4.4 Point

Este paquete está compuesto por la clase Point la cual representa el mínimo elemento en el espacio cartesiano, los puntos. Está compuesto por un vector de tres números en coma flotante, cada uno de ellos representa el valor en los ejes que tiene el punto.

0(x) 1(y) 2(Z)

La clase esta principalmente compuesta por operadores, además tiene un método getter para acceder a cualquiera de las tres componentes del punto y un método de traducción que se

explicará más adelante.

A continuación se explica la lógica que sigue las comparaciones entre puntos.

Se considera que un punto A es mayor que otro B si se cumple lo siguiente:

- La componente Z de A es mayor que la de B.
- La componente Y de A es mayor que la de B.
- La componente X de A es mayor que la de B.
- Las 3 componentes de A son distintas, cada una de ellas, de las de B.

El orden de comparación importa. Es decir, si la componente de Z es menor, se considera que B es mayor que A. En caso de que sean iguales se pasaría a revisar la siguiente componente hasta llegar a la X.

Ejemplos:

- A(27, 12, 0) es menor que B(1, 3, 4)
- A(-4, 12, 0) es mayor que B(1, 3, 0)
- A(0, 8, 7) es menor que B(1, 8, 7)

Dos puntos son distintos si al menos una de las componentes de ambos puntos es distinta. Si las coordenadas de los puntos son iguales X a X, Y a Y y Z a Z significa que son iguales.

Estos operadores tienen gran utilidad en varias partes del código. Ya sea para ordenar los puntos que componen una recta para hacer que la impresora imprima la recta desde begin hasta end, en la intersección entre planos y triángulos, etc.

El método de traducción mencionado al comienzo de esta subsección se llama `translateCoords`. Este es un método que lo que hace es traducir las coordenadas de los puntos a pasos reales que puede dar el motor.

Supongamos un punto A(54.342, 1.737, -15) y que `ADVANCE-PER-ROTATION` es igual a 0,125664:

Si dividimos cada una de sus componentes entre el avance lineal que la impresora hace a cada paso de motor que da(`ADVANCE-PER-ROTATION`), obtenemos el número de pasos, desde el origen de coordenadas, que debe dar para llegar a esas coordenadas en cuestión. Dado

que la división entre los valores puede dar valores en coma flotante y el motor no puede dar pasos decimales, se redondea el valor al entero más cercano. De modo que la traducción del punto A anterior sería (432, 14, -199).

5.1.2.4.5 Triangle

Este paquete está compuesto por la clase Triangle la cual está compuesta por tres instancias de la clase Line, las cuales son los tres lados que componen al triángulo, una instancia de la clase Plane el con plano que contiene al triángulo y una lista de tres puntos que son los vértices que delimitan el ámbito del triángulo.

Esta clase contiene una serie de métodos que son los que llevarán a cabo la intersección entre el propio triángulo y los planos de slice. Esta intersección comienza con una llamada al método intersect. Este método recibe como parámetro el plano a intersectar con el triángulo. El plano pasado por parámetro a esta función siempre será paralelo al eje XY.

Dentro del método distinguimos entre dos casos de intersección. Por un lado si el triángulo está contenido dentro del plano habrá que hacer una intersección especial entre el plano y el triángulo, en caso de que no sea así, se hará otro tipo de intersección.

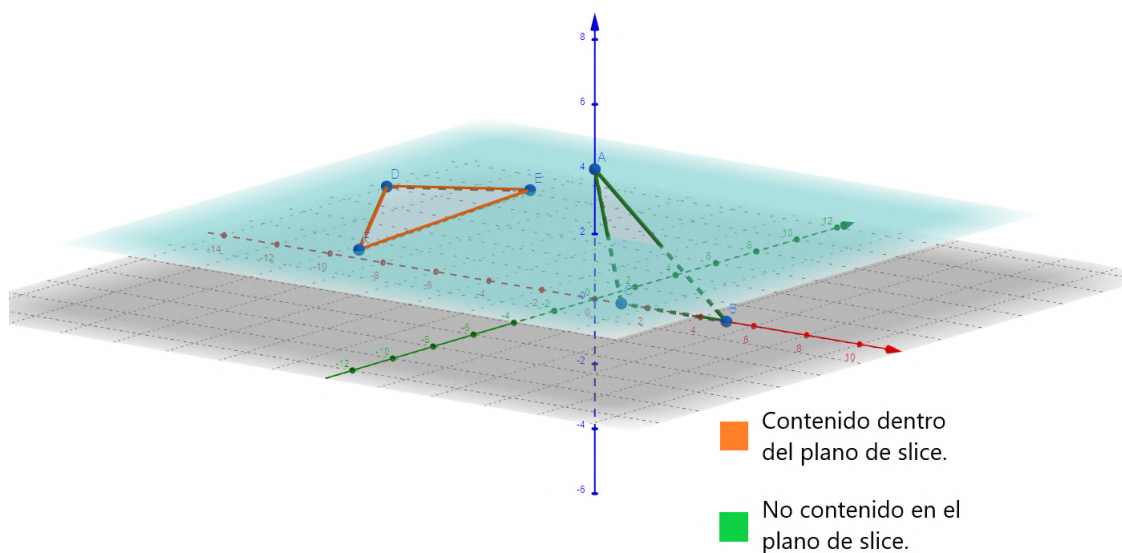


Figura 5.10: Casos de posición relativa entre plano y triángulo.

En el caso de que el triángulo no esté contenido, el método para calcular la intersección es

el siguiente:

- Calculamos la intersección entre el plano que contiene al triángulo y el plano de slice. De lo cual obtenemos una recta de intersección 5.11.

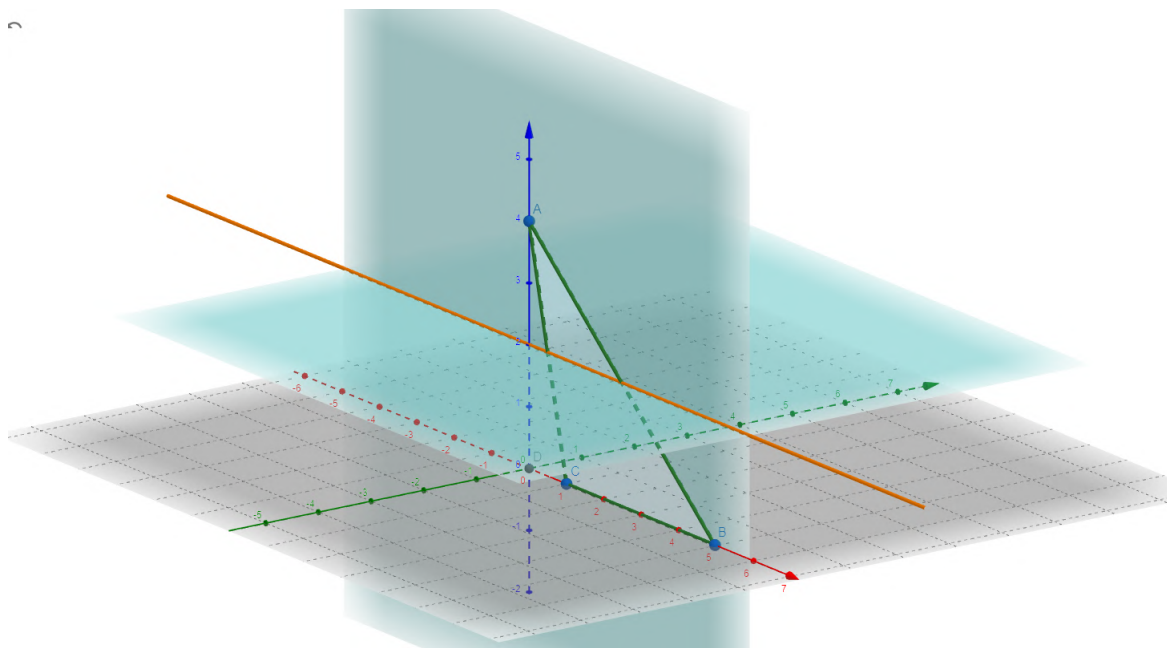


Figura 5.11: Paso 1: Intersección normal entre plano de slice y triángulo.

- Intersectamos dicha recta de intersección con las tres rectas del triángulo. Aquí obtendremos un par de puntos que serán los que conformen la recta de intersección definitiva 5.12.

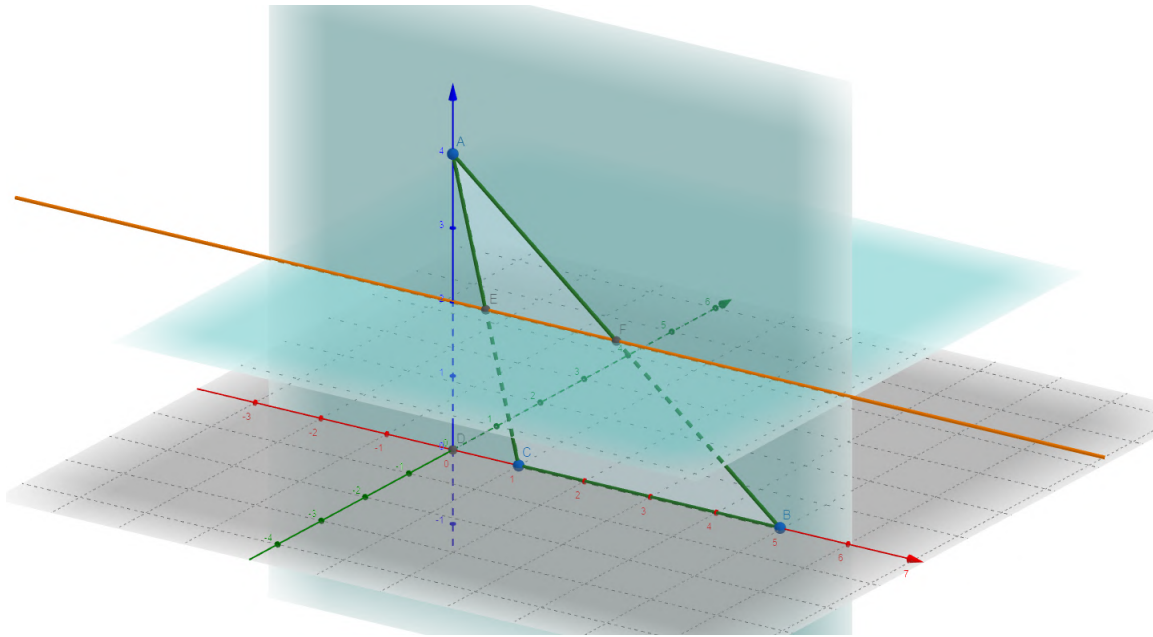


Figura 5.12: Paso 2: Intersección normal entre plano de slice y triángulo.

- Calculamos los puntos que componen esa recta 5.13.

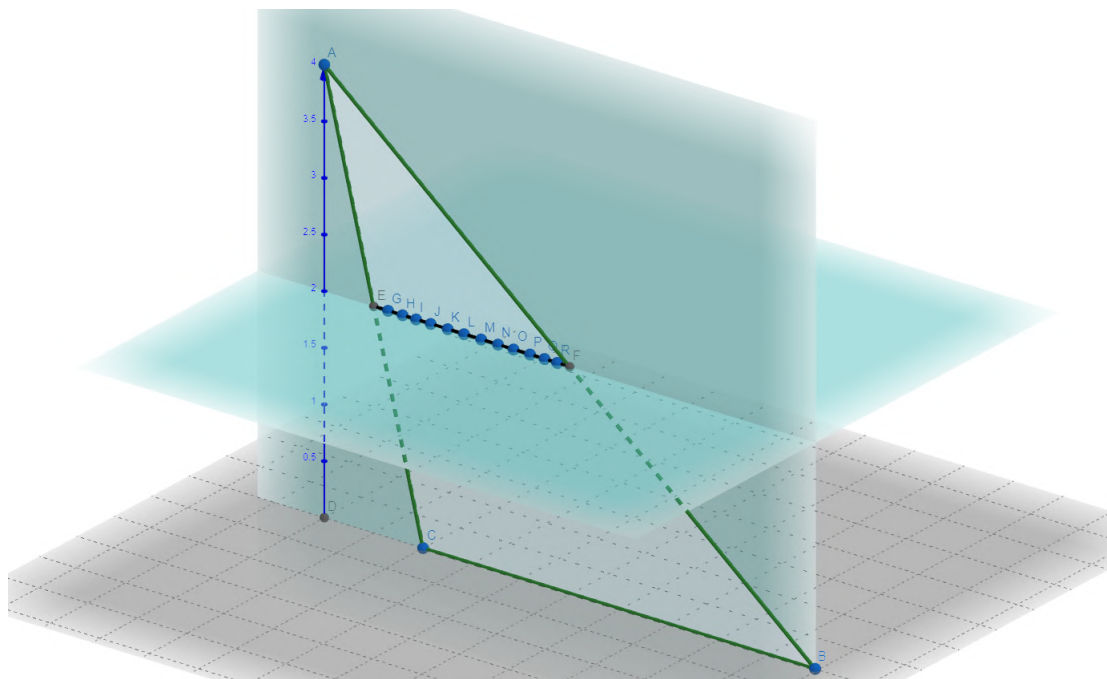


Figura 5.13: Paso 3: Intersección normal entre plano de slice y triángulo.

En el caso de que el triángulo esté contenido, el método para calcular la intersección es el siguiente:

- Calculamos el punto que está contenido en el segmento más largo del triángulo y que simultáneamente pertenece a dicho segmento y a la altura del triángulo, a este punto lo denominaremos el punto límite. Más adelante veremos su utilidad 5.14.

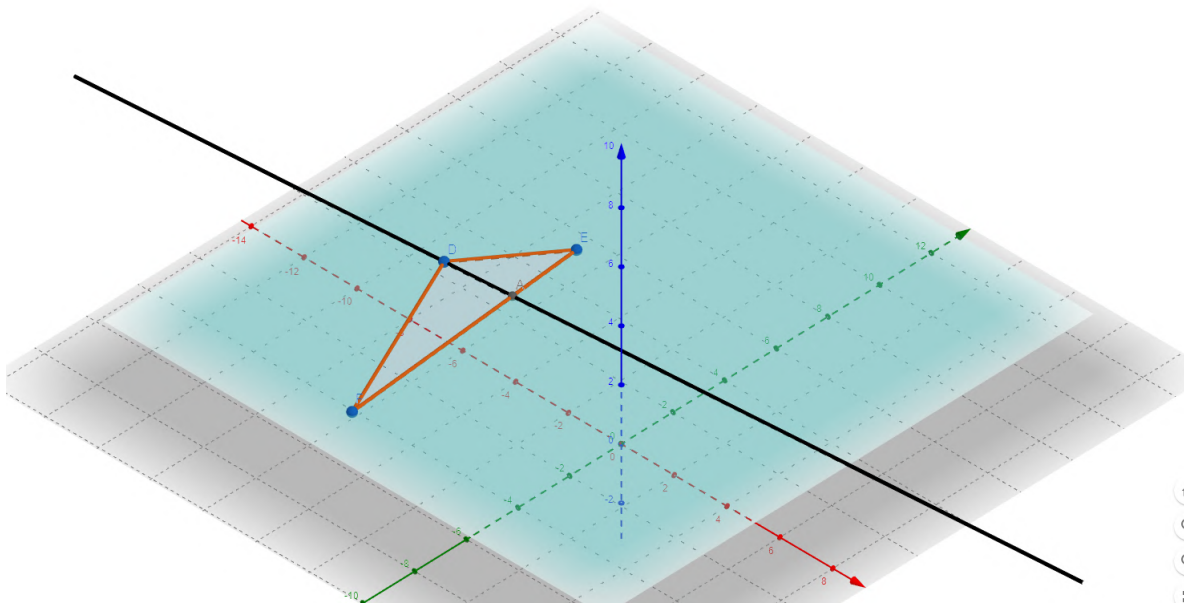


Figura 5.14: Paso 1: Intersección especial entre plano de slice y triángulo.

- Calculamos todos los puntos que componen a la recta que mida más del triángulo 5.15.

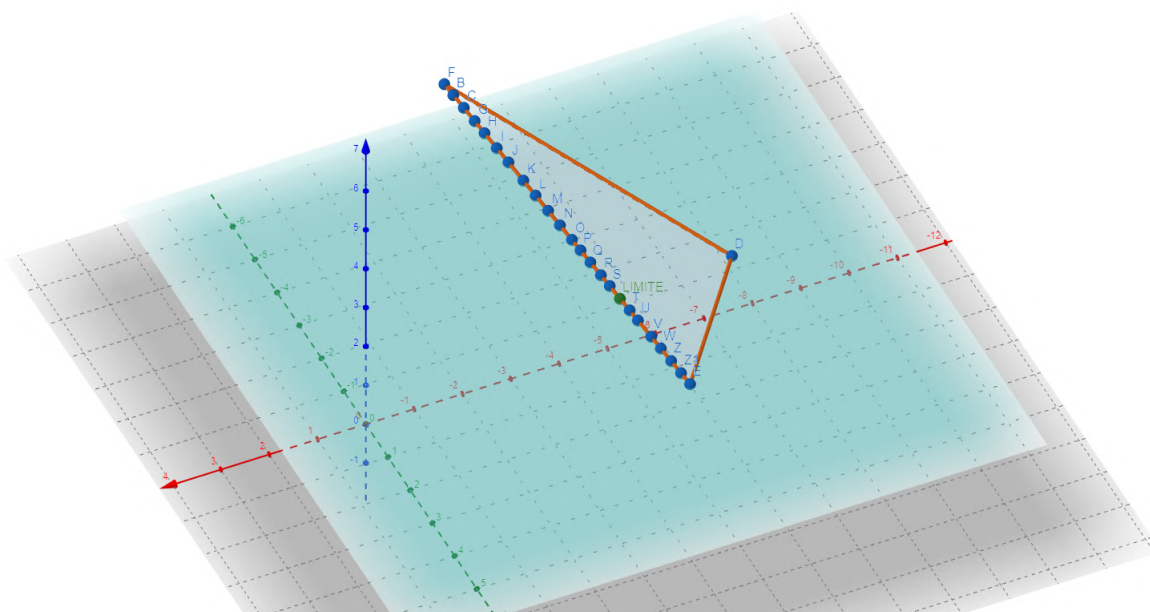


Figura 5.15: Paso 2: Intersección especial entre plano de slice y triángulo.

- Por cada punto obtenido en el paso anterior, creamos un plano perpendicular al punto y a la recta que pasa por ese punto y es perpendicular al segmento de mayor tamaño del triángulo. Luego, intersectamos dicho plano con el plano del triángulo, lo cual nos da una recta. Esa recta la debemos intersectar con alguno de los dos lados restantes del triángulo. Es aquí donde tiene gran utilidad el punto límite. Si el punto perteneciente al segmento más grande del triángulo está a la izquierda del punto límite significa que debemos intersectarlo con la recta FD, en caso contrario con la recta DE 5.16.

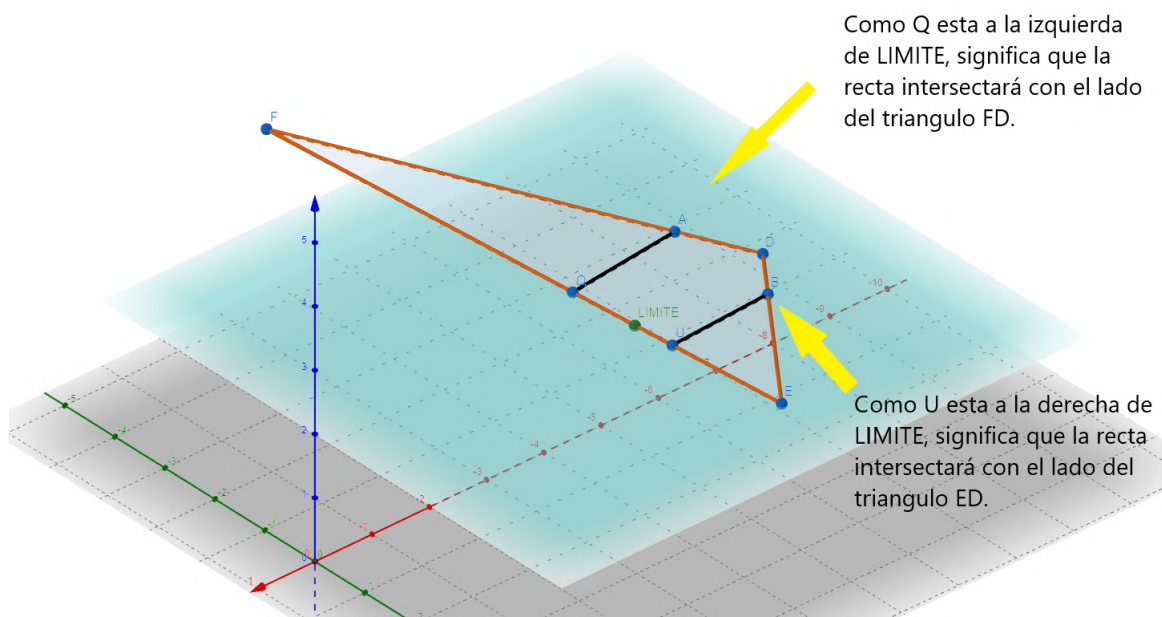


Figura 5.16: Paso 3: Intersección especial entre plano de slice y triángulo.

- Igual que en la intersección normal, una vez tenemos los dos puntos calculamos los puntos de la recta.

5.1.2.4.6 Vector

Este paquete contiene una clase llamada Vector diseñada para la representación de los vectores en el espacio tridimensional.

Esta clase esta compuesta por un vector de 3 componentes, cada componente representa el valor que tiene dicho vector en cada eje coordenado.

0(x) 1(y) 2(z)

La clase contiene varios constructores y un método getter para acceder a cada uno de las componentes del vector.

5.1.2.5. Main

Este paquete contiene un .cpp que se llama Main. Este archivo de código es el punto de entrada al Slicer. En el encontramos una función llamada LoadSTL la cual carga un archivo stl.

La segunda función se llama main y en ella iniciamos todo el proceso de llamadas a Slicer y una vez finalizado, ordenamos las trayectorias obtenidas.

5.1.2.6. Math

Esta paquete contiene una clase llamada Float. Esta clase implementa un tipo en coma flotante. Esta basada en la librería de mpfr [47], desarrollada por GNU que mejora la precisión de las operaciones y comparaciones entre números de coma flotante. También permite asignar más memoria a éstos para que puedan representar números con más dígitos.

Esta clase contiene los operadores básicos para comparar y operar entre ellos.

Todos las variables relativas a los objetos tridimensionales (puntos, rectas y planos) que componen o ayudan a componer la figura, están representados con este tipo de variables. Los motivos principales de usar esta implementación de números en coma flotante y no los tipos estándar proporcionados por C++ son dos:

- Las figuras, dependiendo de su complejidad, pueden tener valores coordenados muy inferiores a los valores límite que nos proporcionan los tipos estándar de coma flotante de C++. El uso de estos tipos nuevos creados garantiza que no se desbordan los límites de las variables.
- La precisión en las operaciones entre números flotantes resulta baja. Esto es debido a la manera en que se guardan los valores en la memoria. De modo, que al utilizar tipos de coma flotante nativos de C++ obtendríamos una mayor imprecisión y variabilidad a la hora de realizar operaciones. Por ejemplo $3.5678 + 6.4321$, dependiendo de una serie de factores, unas veces nos devolvería un resultado y otras veces otro. Con los nuevos tipos que se han implementado, este error se controla en unos límites permisibles de imprecisión.

5.2. Nao

Este software es una interfaz a través de la cual el usuario tiene control de las distintas impresoras que estén en su red local. Con él se pueden enviar peticiones de impresión, analizar el estado de las impresoras, pausar, cancelar, reanudar o abortar la impresión actual, añadir una impresora o eliminarla de la lista de impresoras añadidas.

El flujo normal de trabajo que seguiría es el siguiente:

- El usuario abre la aplicación.
- Añade la impresora/impresoras que imprimirán las figuras.
- Carga una figura con formato .stl en el visualizador de Nao.
- El usuario ordena que una impresora que esté libre imprima la figura cargada y la impresora pasa a estar en estado ocupado.
- Se iría viendo como el porcentaje de impresión va aumentando.
- Si todo salió bien y no se canceló la impresión, al terminar la impresión la impresora le notificaría que ha acabado.

Este software esta implementado en java por dos motivos:

- Contiene herramientas nativas muy poderosas para llevar a cabo una interfaz útil y sin requerir una inversión de tiempo grande en la codificación de ésta.
- La orientación a objetos proporciona facilidad y eficiencia a la hora de implementar las vistas y permite la aplicación del patrón modelo-vista-controlador el cual es muy recomendable para aplicaciones de interfaz.

5.2.1. ¿Dónde opera?

Dentro del esquema general del sistema Nao operaría en la parte del cliente, es decir, en el PC del usuario. Este sería el punto inicial de todo el proceso de comunicación con las impresoras existentes en la red local del usuario. A través de el se inician peticiones de impresión y demás posibles acciones mencionadas anteriormente.

5.2.2. Estructura y composición

La estructura de este software sigue el patrón modelo-vista-controlador. Este patrón consiste en un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. A continuación se expondrán con más en detalle cada una de estos.

Nao está compuesto por un total de 5 paquetes:

5.2.2.1. Common

El paquete common contiene el conjunto de recursos comunes a todas las clases del proyecto. Dentro de esta encontramos 2 subpaquetes:

- **exceptions** Este paquete contiene todos los tipos de excepciones que se pueden producir durante la ejecución del programa así como las excepciones producidas por la parte servidora, la impresora. En este paquete encontramos tres subpaquetes.
 - **connection** Excepciones producidas por problemas relacionados con la conexión.
 - **files** Excepciones producidas por problemas relacionados con el sistema de ficheros del sistema.
 - **scripts** Excepciones producidas tras ejecutar un script y obtener un valor de retorno no esperado, inválido o advirtiéndolo de un error en la ejecución del script.
- **paths** Contiene todas las rutas relativas a la posición del .jar a los recursos como archivos, imágenes, etc usados por las clases del proyecto.

Además este paquete contiene una clase llamada **command**. Esta es una clase que sirve para lanzar scripts en procesos paralelos a la ejecución de la interfaz. En ella de momento solo están implementados métodos para ejecutar scripts en sistemas operativos basados en linux pero está preparado para que en un futuro se puedan implementar en otros sistemas operativos.

5.2.2.2. Controllers

En este paquete están contenidos los controladores, los cuales son los encargados de recibir los eventos de entrada, como modificaciones en las vistas, clicks, etc y también ejecutar las

reglas y la lógica para la gestión de dichos eventos, por ejemplo, si el usuario hace click en un botón estos se encargan de gestionar la lógica que hay detrás de hacer click en ese botón y si fuera pertinente modificar el modelo y advertir a las vistas.

En este paquete encontramos dos clases:

- **ConfigurationController** Este es el encargado de gestionar la comunicación con las impresoras, tanto para enviar como para recibir información de éstas (Petición de estado, petición de impresión, petición de pausa, etc. También de añadir y eliminar impresoras.
- **FigureController** Gestiona los eventos relacionados con la carga de figuras en el visualizador de Nao. Cuando se carga una imagen él gestiona las acciones pertinentes para advertir a la vista de que muestre la imagen en el visualizador y que el modelo guarde la información relativa a esta figura.

5.2.2.3. Main

Este paquete contiene la clase Main, la cual es el punto de entrada al programa. Esta inicializa las vistas y arranca todo el programa.

5.2.2.4. Models

En este paquete están contenidos los modelos, los cuales son los que contienen toda la información que maneja la aplicación Y se encargan de notificar a las vistas los cambios en los datos. Estas modificaciones las notifica gracias al patrón observer.

Estos modelos extienden una clase llamada Observable. Esta les confiere la propiedad de que otras clases puedan observarlas suscribiéndose y de poder notificar modificaciones en los datos de los modelos a los observadores. Cada vez que una vista modifica algún dato esta notifica a sus observadores suscritos que dicho dato ha cambiado. También permite a los observadores desuscribirse.

Dentro del paquete hay dos modelos:

- **ConfigurationInfo** Modelo asociado al controlador ConfigurationController. En este se guarda la información relativa a todas las impresoras guardadas con sus respectivas características y el estado de la conexión del PC del usuario.

- **FigureInfo** Modelo asociado al controlador FigureController. En este se guarda toda la información sobre la figura cargada en el visualizador, sus dimensiones, los triángulos que la componen, etc.

Además de estas dos clases, se encuentra la clase abstracta Observable la cual implementa todos los métodos necesarios para llevar a cabo el patrón observador mencionado anteriormente.

5.2.2.5. Views

Aquí encontramos todas las vistas de la aplicación. Éstas se encargan de recibir datos del modelo y mostrárselos al usuario. Éstas tienen una instancia del controlador asociado a lo que muestra. En este paquete están contenidos los siguiente subpaquetes:

- **Observers** Contiene un par de interfaces de clases que implementan los métodos necesarios para implementar el patrón observador. Esto se complementa con la clase observable, mencionada anteriormente. Los métodos que observen a los modelos tendrán que implementar la clase observer.
- **Principal** Contiene una clase que se llama PrincipalView, ésta es la vista central de la aplicación. Además contiene un paquete llamado elements que es un conjunto de elementos visuales que pueblan la vista principal. Estos son la vista de la figura, FigureView, y el menú principal, MenuBarPanel.
- **Configuration** Contiene las vistas del menú de configuración para añadir o eliminar una impresora así como ver información de estas. En este paquete está la clase Configuration-View que es la vista en sí y un paquete llamado elements que contiene todos los elementos visuales que forman parte de la vista.

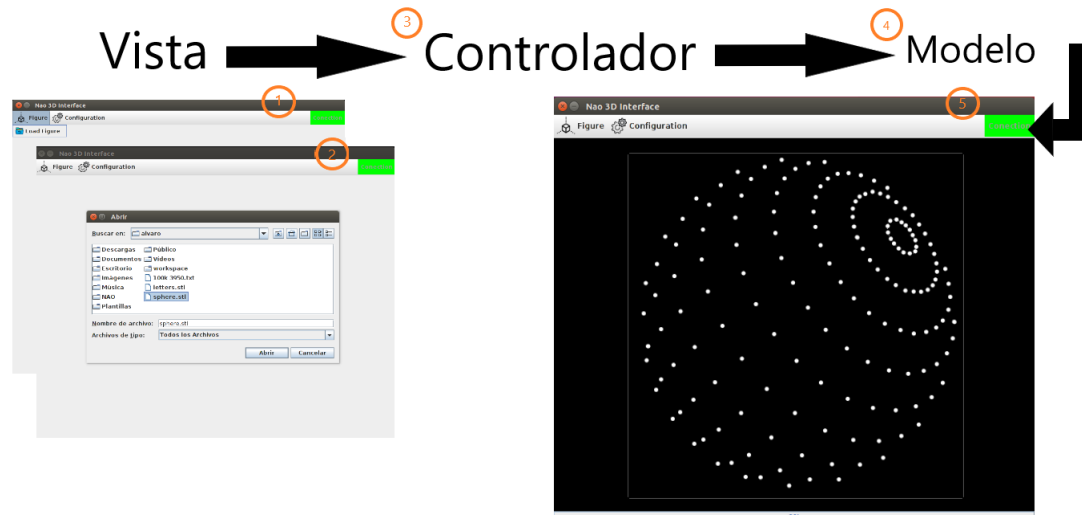


Figura 5.17: Ejemplo del flujo que sigue el patrón modelo-vista-controlador.

- **1 y 2.-** La vista registra la acción del usuario de querer cargar la figura sphere.stl en el visualizador.
- **3.-** El controlador revisa que dicha figura tenga el formato obligatorio .stl y avisa al modelo para que guarde la información.
- **4.-** El modelo guarda la información de todos los triángulos que conforman la figura y avisa a todos los observadores de que se ha cargado una figura.
- **5.-** La vista, al ser observador del modelo, es notificada de la modificación y con la información de la figura recibida por el modelo se actualiza.

5.2.2.6. Carpetas

Además del software mencionado anteriormente, el proyecto está compuesto por una serie de carpetas las cuales contienen elementos externos al código para el funcionamiento de éste. A continuación se exponen y explican sus funcionalidades:

- **Config** Contiene los archivos de configuración de cada impresora añadida. Cada impresora tiene una serie de características únicas, como las medidas de impresión, el puerto de comunicación, y demás elementos que la diferencian del resto.
- **Figure** Carpeta auxiliar donde se guarda la figura en formato .stl y donde posteriormente se guardarán las trayectorias obtenidas de dicha figura al ejecutar el Slicer.

- **FigureMaker** Carpeta donde esta guardado el binario del programa FigureMaker.
- **Img** Contiene todas las imágenes usadas en la interfaz así como un archivo donde se menciona la autoría de cada una de estas imágenes.
- **Scripts** Contiene todos los scripts que usa la interfaz. En ella encontramos:
 - **Config** Carpeta que contiene todos los scripts relacionados con la configuración:
 - **FindPatterns** Dado un patrón encuentra los caracteres que le preceden. Éste sirve para parsear los archivos de configuración de las impresoras y los archivos con formato stl de manera más eficaz y rápida.
 - **GetConfiguracionsFiles** Devuelve la lista de archivos de configuración de impresoras que tiene guardado el programa.
 - **GetProperties** Parsea el archivo de configuración de una impresora en concreto.
 - **PrepareFigure** Es el encargado de ejecutar el binario de Slicer para generar todas las trayectorias que componen la figura.
 - **RemovePrinter** Elimina el archivo de configuración de la impresora seleccionada.
 - **Connection** Contiene todos los scripts relacionados con la comunicación entre el programa y las impresoras:
 - **AddPrinter** Busca la impresora en la red local del PC del usuario.
 - **FindHost** Busca y enumera todos los hosts conectados a la red local del usuario.
 - **SendPetition** Envía una petición a una impresora en concreto.
 - **Env** Script que contiene el conjunto de variables y funciones comunes a todos los scripts mencionados anteriormente. Éste se ejecuta al comienzo del arranque del programa para cargar las variables de entorno.

5.2.3. Mapa de acciones

A continuación se enumeran todas las acciones posibles y las vistas intermedias entre el inicio de cada acción y su desenlace.

5.2.3.1. Cargar una figura

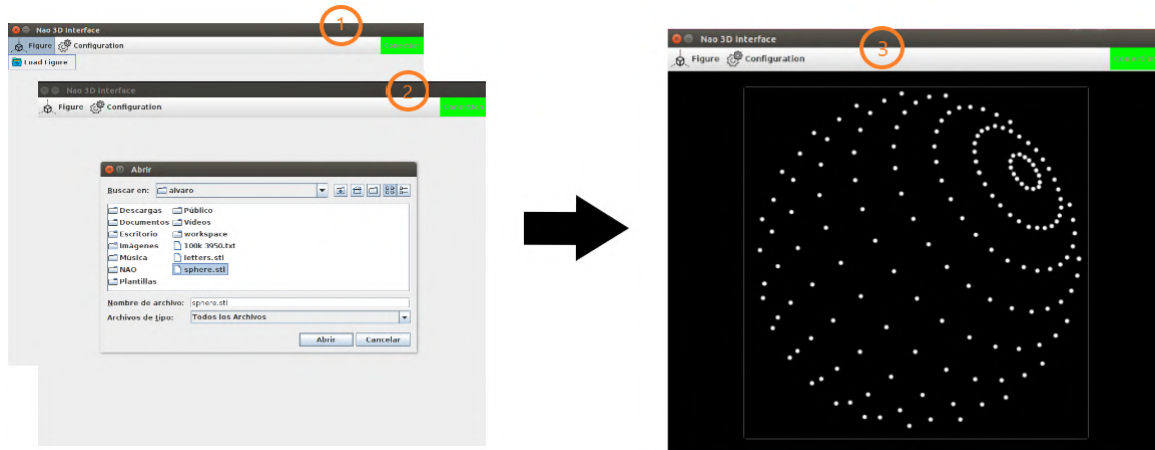


Figura 5.18: Proceso de carga de una figura.

- **1.-** Pulsamos sobre el menú Figure, al hacerlo se desplegará la pestaña Load Figure.
- **2.-** Al pulsar la pestaña Load Figure esta nos abre el explorador de archivos y navegamos hasta la figura que deseemos cargar. Una vez encontrada, pulsamos en abrir. Si se elige algún otro archivo que no tenga extensión .stl o que no tenga el formato esperado para un archivo .stl se advertirá del fallo y se anulará la acción.
- **3.-** Al pulsar en abrir, la imagen será cargada en el visualizador y se mostrará. Si mandáramos imprimir la orden de imprimir a una impresora sería esta figura la que se imprimiría.

En caso de que queramos cambiar la figura solo debemos repetir el mismo proceso que se ha descrito anteriormente seleccionando un archivo distinto al seleccionado anteriormente.

5.2.3.2. Añadir y eliminar una impresora



Figura 5.19: Proceso para añadir una impresora.

- 1.- Pulsamos sobre el menú Configuration, al hacerlo se desplegará la pestaña options.
- 2.- Al pulsar la pestaña options nos abrirá la vista de configuración en la cual tendremos las opciones de añadir impresora, eliminar impresora y abajo se mostrará la lista de impresoras añadidas.
- 3.- Pulsamos la opción de añadir impresora y se nos mostrará un panel de búsqueda en donde debemos introducir el nombre del PC de la impresora, el nombre del usuario a través del cual se realizarán las futuras impresiones y la contraseña del usuario.
- 4.- Rellenamos todos los campos y le damos a al botón buscar. Esto hará que emerge una figura en forma de pacman y comenzará la búsqueda de la impresora en la red.
- 5.- En caso de que no se encuentre la impresora, se notificará a través del panel de búsqueda y este mensaje junto con el pacman desaparecerá
- 6.- En caso de encontrarla, nos lo notificará y este mensaje junto con el pacman desaparecerán.
- 7.- En caso de que se encuentre la impresora, si volvemos al menú de configuración tendremos añadida la impresora en la lista de impresoras. Si presionamos unos segundos sobre el nombre de esta, aparecerán todas las características de la impresora.

Para eliminar una impresora debemos volver al menú de opciones y seleccionar la impresora que deseamos eliminar y pulsar el botón eliminar.

5.2.3.3. Imprimir

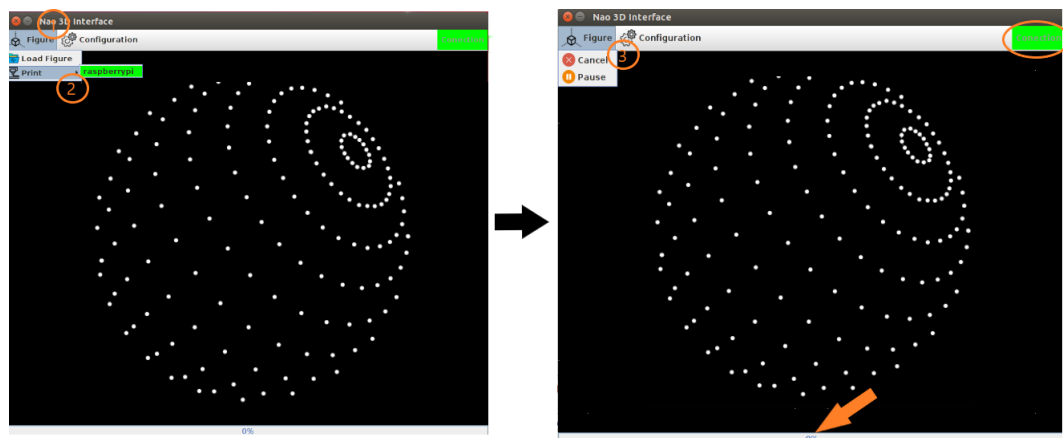


Figura 5.20: Proceso para imprimir una figura.

- **1.-** Una vez cargada una figura en el visualizador y habiendo añadido al menos una impresora a la lista de impresoras conocidas, nos aparecerá una opción nueva en el menú Figura, ésta es la de imprimir.
- **2.-** Si pulsamos en la opción de imprimir nos desplegará la lista de impresoras conocidas. Cada una de éstas tendrá un color, verde si esta disponible, amarilla si esta ocupada y rojo si no se ha encontrado. Solo nos dejará seleccionar una con color verde. De modo que debemos seleccionar una de las impresoras. Si la impresora elegida no tiene el tamaño suficiente de impresión para la figura cargada nos advertirá de que la impresión no se puede realizar.
- **3.-** Una vez seleccionado comenzará la impresión. Si en algún momento deseamos pausar o cancelar la impresión debemos navegar hasta el menú figura de nuevo y allí encontraremos las opciones. Si se pausa la impresión, la opción de reanudación también aparecerá en el menú figura.
- ***** Hay dos elementos informativos que son de gran utilidad durante la impresión. El primero es el estado de la conexión del PC del usuario. Si esta en color verde nos indica que esta conectado a su router, de lo contrario nos lo indicaría con un color rojo. También, durante la impresión se ve un porcentaje debajo de la figura, que indica el porcentaje total de impresión que lleva realizado la impresora.

Cabe destacar que para que sea posible imprimir una figura deben cumplirse las siguientes condiciones:

- Se debe haber cargado una figura previamente en el visualizador.
- Se debe haber añadido una impresora satisfactoriamente previamente
- Al menos una de las impresoras está en estado disponible.
- Los tamaños de las figuras no deben sobrepasar los tamaños máximos de impresión de la impresora.

5.2.4. Comunicación y procesos paralelos

Simultáneamente a la ejecución de la aplicación, se ejecutan otros procesos que de manera auxiliar ayudan al normal funcionamiento de la aplicación. Estos son los encargados de establecer comunicaciones con las impresoras, enviar o recoger información de la impresora, chequear el estado de la impresión y enviar peticiones u órdenes a la impresora. Todas estas comunicaciones se realizan mediante paquetes TCP. La impresora ejecuta un servidor TCP que está a la espera de recepción de mensajes.

A continuación se irán explicando el funcionamiento y el papel desempeñado por estos procesos.

5.2.4.1. CheckStatus

Este se ejecuta desde el comienzo de la ejecución de la aplicación y es el encargado de:

- Revisar que el host donde se ejecuta el programa este conectado a un router. Esto es vital ya que sin conexión local, la aplicación no podría comunicarse de ningún modo con las impresoras.
- Si detecta que esta conectado a una red local, escanea esta y revisa el estado de las impresoras guardadas. Es decir, va preguntando periódicamente a cada una de las impresoras por su estado, y éstas pueden contestarle que están ocupadas o no. También cabe la posibilidad de que no las encuentre de modo que se considera que la impresora tiene un estado inalcanzable.

Cabe destacar que este proceso se detiene cuando el usuario ordena a una de las impresoras imprimir, dejando dormido a este proceso hasta la finalización de la impresión.

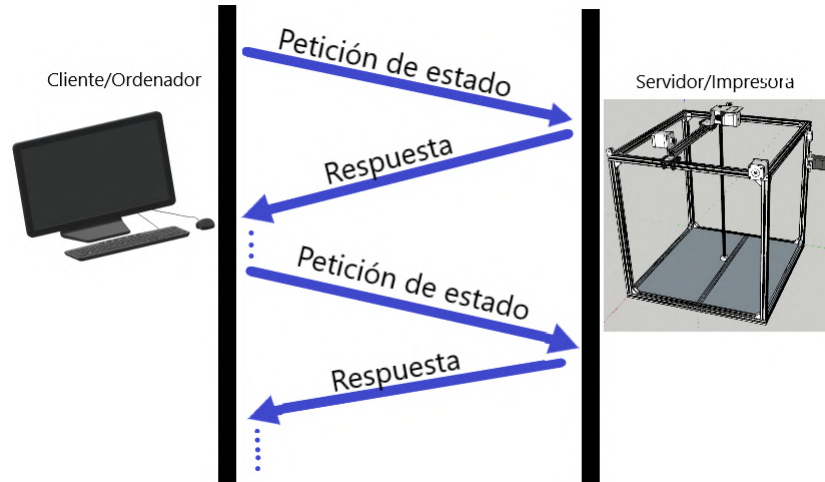


Figura 5.21: Esquema de funcionamiento normal del proceso CheckStatus.

5.2.4.2. CheckInterruptions

Este proceso empieza a actuar cuando el usuario decide ordenar a una impresora que imprima una figura cargada previamente en el visualizador de la aplicación. Se encarga de pedir a la impresora información acerca del estado de la impresión:

- Porcentaje de impresión
- Si se pausó, canceló, reanudó o abortó manualmente la impresión. Ya que si lo recordamos del capítulo de 4.2.8, la placa tiene mecanismos internos para pausar, reanudar o abortar la impresión. Esta es una medida cautelar en caso de que la impresora por el motivo que sea quede incomunicada y haya que hacer una parada de emergencia.

Esta comunicación con el servidor de la impresora utiliza la técnica piggyback, es decir, hace solapamiento de mensajes para reducir el tráfico de paquetes TCP en la red.

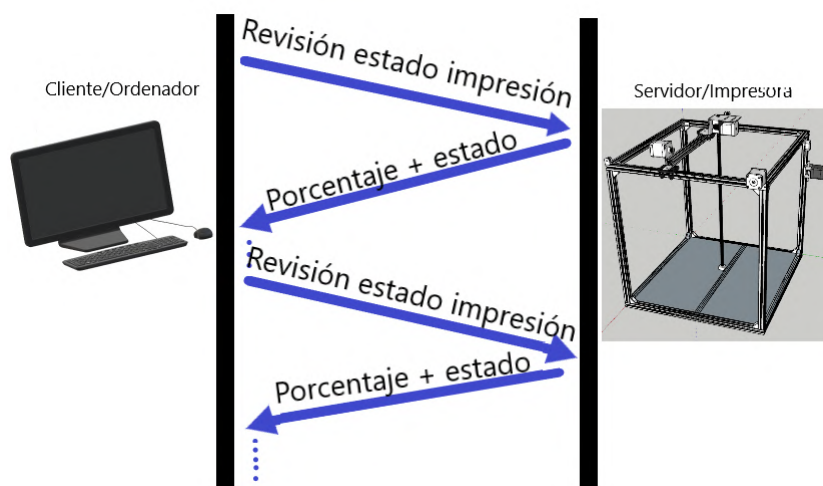


Figura 5.22: Esquema de funcionamiento normal del proceso CheckInterruptions.

Durante la ejecución de la impresión, por el motivo que sea, el usuario puede desear que la impresión se detenga o se aborte, es por ello que la aplicación tiene un mecanismo para tales eventualidades. Cuando el usuario a través de la interfaz ordena a la impresora pausarse se envía una orden a la impresora de que se detenga hasta nuevo aviso y esto provoca que este proceso quede dormido y se despierte al proceso CheckStatus hasta que, o bien se cancele, de modo manual o a través de la aplicación, la impresión se reanude o se cancele de iguales modos que los mencionados anteriormente. Si se canceló este proceso moriría y se volvería a la vista principal, esto también ocurriría si la impresión terminara en circunstancias normales. Si el usuario decide reanudar la impresión este proceso se despertaría y se volvería a dejar durmiendo a CheckStatus. Solo puede estar ejecutándose uno de estos dos procesos, nunca simultáneamente.

5.2.4.3. AddPrinter

Este proceso sirve cuando queremos añadir una impresora existente en nuestra red. Éste, con los datos que identifican a dicha impresora, busca en la red gracias al script FindHosts.sh todos los hosts conectados y si alguno coincide con las credenciales aportadas por el usuario se añadirá a la lista de impresoras conocidas y además se copiará mediante SCP su archivo de configuración el cual está guardado en el sistema de ficheros de la impresora.

5.3. Line printer

Este software, escrito en C++, es el encargado de registrar y procesar todas las peticiones y ordenes que se reciben desde los usuarios. Es el servidor TCP con el que se conectan los usuarios para realizar sus impresiones.

5.3.1. ¿Dónde opera?

Este software operaría en la impresora y comenzaría su ejecución a la par que el arranque del sistema operativo.

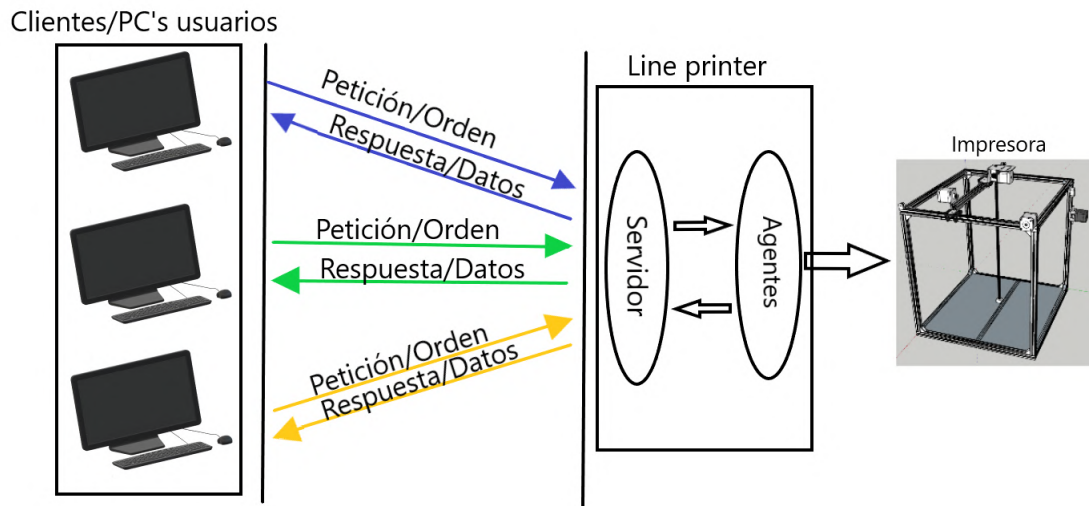


Figura 5.23: Lugar que ocupa Line printer en el esquema general.

Como se muestra en la imagen anterior este software está compuesto por dos partes, dos hilos de ejecución que tienen distintos propósitos. El propósito del servidor es la de atender toda petición/orden recibida por los usuarios y responder a dicha petición o devolver información acerca de la impresión. Este servidor solo actúa ante la petición de un usuario, de modo que este queda en reposo hasta que un usuario envía una petición/orden a este. Por otro lado tendríamos a los agentes, estos son los encargados de llevar la impresión a cabo. Cada agente representa el software de control de cada una de las partes que componen la impresora, motores, termistor, relé, etc. Los agentes se comunican entre ellos para ejecutar la impresión de manera coordinada.

Estos dos hilos se ejecutan bajo el mismo proceso y se comunican a través de pipes para coordinarse en función de las ordenes recibidas por los usuarios o para suministrar información

al usuario del estado de la impresión.

5.3.2. Estructura y composición

Como se menciono anteriormente, este software está compuesto por una parte servidora y por los agentes que llevan a cabo la impresión. Para mayor claridad se explicará por separado cada una de estas partes pero hay que recordar que estos no son dos procesos distintos, son dos hilos que forman parte del mismo proceso.

En la raíz del proyecto encontramos los siguientes elementos:

- Paquete **Agents**
- Paquete **Server**
- Un archivo de código llamado **Main**. Esta clase es el punto de inicio del programa. Aquí se arranca el servidor.
- Un archivo de cabecera **ReturnsEnum**. En este archivo están contenidos todos los tipos de devolución que puede responder el servidor a los clientes.

En el paquete Server se encuentra la clase Server, la cual tiene implementados métodos para llevar a cabo la comunicación con los clientes. De esta implementación cabe destacar el método `parsePetition` y el método `addLogEntry`.

El servidor está a la espera de mensajes y cuando recibe un mensaje lo analiza y emite una respuesta. Los mensajes que vienen de los usuarios son números que representan que es lo que está pidiendo el usuario. El código de estas peticiones y su representación numérica esta registrado en el enumerado `petitions` el cual está implementado en el archivo de cabecera de la clase Server. De modo que cuando se recibe un mensaje, este va acompañado de uno de estos códigos y el método `parsePetition` se encarga de ejecutar una acción en función de dicho código. Si el código no corresponde con ninguna acción conocida o si la acción en sí no tiene sentido se envía un mensaje al usuario notificando el error.

Cada vez que un usuario interactúa con el servidor se guarda la actividad realizada, la IP del PC del usuario que ha realizado la petición y la hora a la que se hizo en un archivo de registro. Para llevar a cabo esta acción de registro se usa el método `addLogEntry`. El servidor seguirá

ejecutándose hasta que se apague el sistema o se produzca un error a la hora de registrar una actividad en el archivo de registro en cuyo caso saldrá una ventana emergente en el sistema operativo de la impresora indicando cuál ha sido el error así como una entrada en el archivo de log que indica el motivo del error y se cerrará el servidor.

Por otro lado tendríamos el paquete Agents, el cual está compuesto por los siguientes elementos:

5.3.2.1. Clase Agent

Es la implementación de la clase base de todos los agentes. En ella se encuentran todos los mecanismos que deben tener los agentes para comunicarse entre ellos así como una implementación de un manejador de señales el cual sirve para pausar, reanudar o cancelar la impresión. Cuando se levanta la señal de SIGABRT o SIGINT este método es ejecutado en cada una de las clases que lo implementen. La señal SIGABRT es usada para implementar la cancelación de la impresión y SIGINT para implementar la pausa/reanudación de impresión.

5.3.2.2. Paquete Interrupt

Dentro de este se encuentra la clase Interrupt, un tipo de agente encargado de revisar si se produjo una pausa, cancelación o reanudación de la impresión. En el caso de que se produzca alguna de estas este agente hará levantar las señales SIGABRT en caso de cancelación o SIGINT en caso de pausa, lo cual hará que el resto de agentes ejecuten sus implementaciones de los manejadores de señales. En el caso de pausa, deja a los agentes dormidos hasta que el agente interrupt recibe la orden de reanudar ya sea manualmente o vía Nao y éste libera la variable condicional haciendo que todos los agentes se reactiven.

Este agente está comunicado directamente con el servidor. Le comunicará al servidor si se hizo una pausa/cancelación/reanudación manual y el servidor le comunicará los mismo pero vía Nao. La comunicación entre estos dos hilos es mediante pipes anónimos.

5.3.2.3. Paquete Thermistor

En este paquete se encuentra la clase Thermistor que implementa un agente encargado de controlar la temperatura del extrusor y la medición de temperatura. Cuando se instancia el objeto, el agente recibe una serie de parámetros. Entre esos parámetros se encuentran un par de

valores que le indican que rango de temperaturas en grados son los óptimos para la impresión, si se sale de estos rangos apaga o enciende el extrusor a través del pin de control del relé.

Con el termistor de la placa es con el que obtiene la temperatura, pero para hacer las lecturas de temperatura debemos hacer ciertas conversiones de valores. Esta parte es la continuación de la explicación de la sección 4.2.4 en la cual se explicaba el funcionamiento del conversor analogico-digital MCP3002, la figura 5.24 es una simplificación de ésta sección.

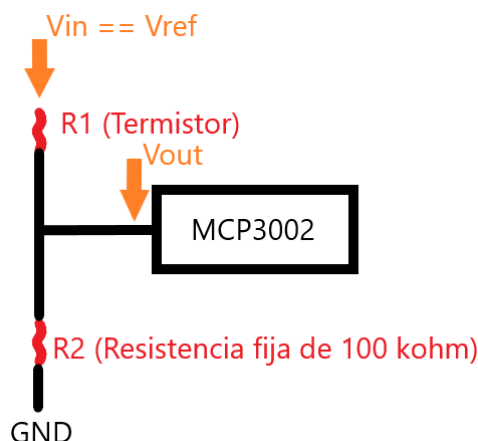


Figura 5.24: Simplificación del circuito de lectura de temperaturas.

A nosotros lo que nos interesa realmente es averiguar el valor de la resistencia $R1$ del termistor. Si recordamos de la sección 4.2.6, el fabricante nos proporciona una tabla en la que se relaciona la resistencia que tiene el termistor y la temperatura, de modo que para calcular esa resistencia primero debemos obtener la fórmula que describe los valores leídos por el MCP3002. Esta relación de valores resistencia-temperatura será guardado en un mapa en el que la resistencia será la clave y la temperatura el valor.

Cuando se ordena al MCP3002 que lea la temperatura nos devuelve un valor entre 0 y 1023 que representa el voltaje recibido por el dispositivo. El rango de valores comprende entre 0 y 1023 ya que el tamaño de registro que tiene el MCP3002 es de 10 bits. Como el voltaje es una medida analógica, debemos hacer un ajuste para convertirlo a una digital y que la placa controladora pueda operar con este valor leído. De modo que lo que hacemos es una regla de tres diciendo que: Si el valor devuelto por el MCP3002 es 1023 significa que está recibiendo V_{ref} , es decir, V_{ref} es el voltaje de referencia que le estamos entregando y es el máximo posible que puede recibir. De modo que al simplificar la regla de tres obtenemos:

$$V_{out} = \frac{ret * V_{ref}}{1024}$$

Siendo V_{out} los voltios recibidos por el MCP3002, ret el valor comprendido entre 0 y 1023 devuelto por el MCP3002 y V_{ref} el voltaje de referencia que tiene el MCP3002.

Ahora, aplicando la ley de ohm al circuito que tenemos en la figura 5.24 nos quedaría que:

$$R1 = \left(\frac{V_{in}}{V_{out}} - 1 \right) * R2$$

Si sustituimos la primera ecuación dentro de la segunda obtenemos que:

$$R1 = \left(\frac{1024}{ret} - 1 \right) * R2$$

Recordemos que ret es el valor, comprendido entre 0 y 1023, leído por el MCP3002 y $R2$ es la resistencia fija que esta en la placa que equivale a 100 kohm. De modo que una vez que leemos el valor devuelto por el MCP3002 podemos aplicar esta última fórmula, la cual nos dará la resistencia que está ejerciendo el termistor, $R1$, en el momento de la lectura. Con este valor buscamos en el mapa mencionado anteriormente y así es como se obtiene la temperatura actual.

Una cosa importante que cabe destacar es que las lecturas pueden ser imprecisas, es decir, a veces hay ruido en el termistor producido por picos de intensidad en el circuito. Es por ello que para prevenir las lecturas de temperaturas erróneas, en vez de leer una temperatura e interpretarla como la verdadera, se toman varias temperaturas de modo que se analicen varias simultáneamente del mismo momento y comparándolas entre sí podemos afinar mejor la temperatura.

5.3.2.4. Paquete Stepper

El paquete contiene la clase Stepper, esta clase es la implementación del software de control de los motores bipolares. En la ejecución de una impresión habrá cuatro instancias de este agente, una para el eje x, otra para el y, otra para el z y otra para el motor del extrusor.

Estos agentes guardan sus posiciones, en pasos de motor, en todo momento para saber donde están y a donde deben dirigirse cuando se les ordene moverse.

5.3.2.5. Paquete Boss

Este paquete contiene la clase Boss, la cual es la implementación del agente Boss, un agente que se encarga de coordinar y ordenar actuar a todos los demás agentes y, además, mantiene comunicaciones con el servidor.

Este agente va leyendo los puntos que componen a la figura para ordenar y coordinar a los agentes Stepper, los cuales irán siguiendo las instrucciones del agente Boss para ir imprimiendo la figura, además activa a los agentes Interrupt y Thermistor para que vayan trabajando paralelamente al procesamiento de puntos.

Este agente va comunicando al servidor el porcentaje de líneas impresas de la figura a través de una variable compartida entre threads.

5.3.3. Carpetas

Este software esta embebido en una carpeta llamada AlTu la cual tiene la siguiente estructura:

- **bin** Contiene el binario del Line printer
- **conf** Guarda un par de archivos de configuración, las medidas de los tiempo están en micro segundos, las medidas de tamaños en milímetros y las de temperaturas en grados centigrados:
 - **characteristics** Archivo que contiene los pines de control y demás parametros necesarios para cada elemento hardware de la impresora. Estos los reciben los agentes como argumentos de entrada cuando se instancian.
 - **configuration** Archivo que contiene los datos característicos de la impresora.
- **figure** Guarda el conjunto de trayectorias que componen la figura que va a ser impresa. Cuando un usuario desea imprimir una figura, se llamará al Slicer para crear este documento y posteriormente, el documento resultante se copiará en esta carpeta.
- **logs** Carpeta que contiene el archivo de registro de la impresora.
- **scripts** Carpeta que contiene el script de arranque del Line printer. Este se encarga de asegurar que toda la instalación está bien y que todo está en las rutas correctas, obtener cuál es la IP de la impresora y arrancar el servidor. Si se detecta que no hay conexión con el router este script intentará reconectarse hasta que lo consiga. Si por el motivo que sea la escritura en el archivo de registros falla este script terminará y dejará constancia del fallo en el archivo de registros y mostrará una ventana emergente en el sistema operativo.

16 12 500	Motor eje X => (step, dir y tiempo entre pasos)
21 20 500	Motor eje y => (step, dir y tiempo entre pasos)
23 18 500	Motor eje z => (step, dir y tiempo entre pasos)
14 15 500	Motor extrusor => (step, dir y tiempo entre pasos)
100 7 230 190 5000000	Termistor y relé (canal SPI, pin relé, max temp, min temp, tiempo entre lecturas)
25 24 6 13 26 19 5000000	Botones de interrupción(pin abrt1, pin abrt2, pin pause1, pin pause2, pin detectFilamento1, pin detectFilamento2, tiempo entre lecturas)
1900 1900 1900	Coordenadas del centro de la impresora medidas en pasos (x, y, z)

Figura 5.25: Formato del archivo characteristics.

```

name: raspberrypi Nombre de host de la impresora
user: pi Usuario de la impresora
aprX: 0.1
aprY: 0.1 Avance por rotación de cada motor de eje
aprZ: 0.1
X: 380
Y: 380 Dimensiones de impresión
Z: 380
p: 8888 Puerto del servidor TCP
ed: 0.4 Diametro de extrusión del filamento

```

Figura 5.26: Formato del archivo configuration.

```

[ 08/03/2019 17:06:28 ] Status request from: 192.168.1.43
[ 08/03/2019 19:16:24 ] Status request from: 192.168.1.43
[ 08/03/2019 19:16:32 ] Request to print agreed to: 192.168.1.43
[ 08/03/2019 19:16:43 ] Sent Figure from: 192.168.1.43
[ 08/03/2019 19:16:43 ] The printing begins : 192.168.1.43
[ 08/03/2019 19:16:50 ] Returns print percentage to : 192.168.1.43
[ 08/03/2019 19:17:02 ] Check interruptions petition from : 192.168.1.43
[ 08/03/2019 19:17:16 ] Returns print percentage to : 192.168.1.43
[ 08/03/2019 19:17:29 ] Check interruptions petition from : 192.168.1.43
[ 08/03/2019 19:17:41 ] Returns print percentage to : 192.168.1.43
[ 08/03/2019 19:17:46 ] Check interruptions petition from : 192.168.1.43

```

Figura 5.27: Formato del archivo registro de actividad del servidor.

Capítulo 6

Apartado económico

Componente	Unidades	Precio	Suministrador(enlace)
Perfil de aluminio T-slot 5 20x20x500	13	34,96	MakerShop
Tornillo allen cabeza ultra plana M5	75	7,26	MakerShop
Escuadra unión 20x20 aluminio para T-slot	28	17,08	MakerShop
Polea gt2 20 dientes para eje Ø5mm	4	9,81	MakerShop
Correa GT2 abierta	2	7,6	MakerShop
Escuadra 20x20 aluminio anodizado	2	4,82	MakerShop
Guía lineal/Patín	3	48,84	MakerShop
Motor Mount Plate Nema 17	3	21,78	MakerShop
T-nut con bola resorte	75	21,78	MakerShop
Tubo de teflón para filamento 1.75mm	1	2,42	MakerShop
Placa de unión 20x80 aluminio anodizado	1	3,25	MakerShop
Perfil de sujeción del eje z	1	0	-
Guía cremallera dentada	1	0	-
Cremallera dentada cilíndrica Ø10mm 1000mm	1	0	Norelem
Polea dentada de 17 dientes	1	0	Norelem
Tornillería y tuercas varias	18	3	Bauhaus
Plataforma de impresión	1	13.45	Leroy Merlin
COSTE TOTAL		196,05	

Cuadro 6.1: Lista de materiales de la estructura y sus costes.

Componente	Unidades	Precio	Suministrador(enlace)
DRV8825	4	47,8	Electrónica Embajadores
Detector de filamento	1	4,99	SM SunniMix
Funda Raspberrypi con ventilador integrado	1	14,99	BonRob
Extrusor mk8 pack completo	1	54,65	Moebyus Machines
Relé CMX60D5	1	32,4	Digi-Key
Fuente de alimentación 12V 350W	1	40,3	Electrónica Embajadores
Nema 17 modelo 42byghm809	3	51,63	Diotronic
Ventilador vapo 40x40x6 5V	2	21,6	Diotronic
Raspberry Pi 3 +B	1	41,99	MediaMarkt
Placa PCB de control	1	42,04	JLCPCB
Resistencia 100K 2W	1	0,2	Electrónica Embajadores
Zocalo 8 pines	1	0,07	Electrónica Embajadores
Tornillería varia	36	3	Bauhaus
Diodo 10A	1	0,7	Electrónica Embajadores
MCP3002	1	4,75	Electrónica Embajadores
Condensador electrolítico 25V	4	1,48	Electrónica Embajadores
Cable 4 hilos	3	2,07	Impresoras3dslowcost
Cable 2 hilos	1	0,88	Impresoras3dslowcost
Conectores dupont + Pin macho dupont	40	12,8	Impresoras3dslowcost
Conectores termoretractiles	22	2,63	Hengsheng ES
Varias regletas pin 2,54mm macho	40	10,4	Electrónica Embajadores
Varias regletas pin 2,54mm hembra	40	26	Electrónica Embajadores
COSTE TOTAL		375,4	

Cuadro 6.2: Lista de componentes que comprenden la placa y sus costes.

Partes	Precio
Hardware	375,4
Estructura	196,05
COSTE TOTAL	571,45

Cuadro 6.3: Coste total del proyecto.

Cabe destacar que el precio calculado es solo de los materiales y demás componentes que son necesarios para la construcción definitiva del prototipo actual, no se han tenido en cuenta los gastos requeridos en la fase de I+D, las herramientas necesarias para hacer las modificaciones en las piezas, las averías o reparaciones de ciertos componentes, los traslados necesarios para la compra u obtención de información pertinente o el sobrecargo de encargar piezas y pagar a los transportistas.

También hay que entender que este precio elevado se debe a que estas piezas se compran al por menor, haciendo que individualmente sea mucho más caro, si esto se hiciera a gran escala, el precio total se reduciría drásticamente.

Capítulo 7

Aplicación de conocimientos

Durante el desarrollo del proyecto he tenido la oportunidad de poner a prueba todas mis habilidades y la gran mayoría de conocimientos adquiridos durante la carrera. Esta es la relación entre las asignaturas cursadas y la utilidad que me han aportado durante el proyecto:

- **Métodos matemáticos de la ingeniería y Ampliación de matemáticas** Me ayudo para ha asentar las bases en cálculo y álgebra para realizar los cálculos estructurales de la impresora.
- **Fundamentos de la programación** Me ha ayudado a sentar las bases de la programación para entenderla en un marco más general y poder adaptarme a distintos tipos de lenguajes y contextos en los que aplicar dichos lenguajes.
- **Tecnología de la programación** Me ha ayudado a estructurar de una manera más eficiente el proyecto respetando los patrones y usándolos en los casos pertinentes.
- **Fundamentos de la electricidad y electrónica, Fundamentos de computadores, Tecnología y organización de computadores, Estructura de computadores y Arquitectura de Computadores** Me ha dado los conocimientos básicos para poder realizar toda la parte del diseño hardware.
- **Robótica** Me enseñó el funcionamiento básico de motores paso a paso y electrónica básica para el manejo de estos así como aspectos de geometría computacional.
- **Desarrollo de sistemas interactivos** Me ayudó a desarrollar una interfaz sencilla y usable

que tuviera una curva de aprendizaje sencilla y adaptar dicha interfaz al usuario final sin que ésta esté adaptada a la visión de un ingeniero.

- **Redes, Sistemas operativos y Ampliación de redes y sistemas operativos** Fue fundamental para poder desarrollar todo el protocolo de comunicaciones entre las impresoras y las aplicaciones así como los scripts y demás elementos del sistema que intervienen en la impresión.
- **Estructuras de datos y algoritmos y Métodos algorítmicos en resolución de problemas** Aparte de ayudarme a comprender los conceptos básicos acerca de la rama de las matemáticas de la complejidad de los algoritmos me ayudo a desarrollar un software como el Slicer reduciendo su complejidad, el cual tiene que resolver un problema con un cardinal de entrada muy grande y que sin estos conocimientos podría tener un coste computacional muy elevado y tardaría mucho tiempo de ejecución en resolverlo.
- **Programación concurrente** Los conocimientos de esta asignatura fueron vitales para la implementación del Line printer, de la interfaz Nao y de la interacción entre estos programas.

Capítulo 8

Conclusiones

La impresión 3D es un campo en plena ebullición el cual no para de avanzar día a día, nuevos avances, técnicas de impresión y hallazgos hacen que esta tecnología sea cada vez más poderosa y tenga una mayor versatilidad haciendo que tenga utilidad en prácticamente todos los campos de la sociedad. Parece que es el camino al que apunta el futuro y este parece prometedor.

Durante el desarrollo del proyecto, a medida que avanzaba, me fui percatando de la dificultad que desentraña el desempeño de esta tecnología, son muchos los factores decisivos que determinan que funcione o no la implementación de una impresora y la probabilidad de que surjan inconvenientes es alta. Esto combinado con el poco conocimiento previo en la materia es lo que ha ralentizado con mayor grado el desempeño de este trabajo.

Este proyecto tenía por objetivos:

- Desarrollar mis aptitudes ingenieriles adquiridas durante la carrera y poner a prueba éstas.
- Sentar las bases de un diseño software-hardware para extender esta tecnología.
- Potenciar esta tecnología.

De estos, el primero se cumplió muchas más allá de mis expectativas, el conocimiento adquirido durante el desarrollo de este proyecto ha sido de proporciones muy grandes, tanto en materias relacionadas con la carrera como en otros ámbitos fuera de ésta como es:

- Mejorar notablemente mi capacidad de razonamiento y deducción en la resolución de problemas.

- Ayudar a estructurar mi cabeza de una manera más ordenada y eficiente dando como resultado un proyecto ordenado y lógico que siguiera unos pasos firmes en su desarrollo.
- Mejorar mi capacidad para estimar cuanto tiempo llevaría cada tarea y como de compleja era. Lo cual produjo una mejora en mi capacidad de gestionar los tiempos del proyecto.

En cuanto a los otro dos objetivos podemos decir que sí que se cumplieron pero con ciertas reticencias. Es cierto que he logrado desarrollar la impresora pero por el momento dista de ser una opción plenamente viable a las alternativas actuales. Aun queda mucho trabajo por delante para que ésta sea una opción igual de válida que las alternativas. Estos son los inconvenientes o partes más débiles del proyecto:

- Solo permite imprimir en PLA, no tiene preparado un sistema para imprimir en otros materiales más sólidos y resistentes.
- El eje x tiene ciertos problemas mecánicos, a veces las guías lineales del eje patinan y esto produce errores de precisión en la impresión.
- El eje z baila debido a la poca fricción que hay entre la guía de la cremallera y la cremallera en sí, provocando que cuando hay pequeñas vibraciones desplace el eje y produzca errores de precisión en la impresión.
- El software Slicer no es todo lo eficiente que podría ser, tiene varias mejoras posibles para reducir el coste computacional en su ejecución. Además, no tiene ninguna opción para rellenar automáticamente las figuras según las necesidades del usuario ni generar automáticamente estructuras de soporte para crear figuras voladizas como tienen otros softwares como el Cura o el Slic3r, lo cual lo hace mucho menos potente que sus programas homólogos.
- La placa tiene elementos sobre dimensionados para las necesidades requeridas lo cual representa un sobre coste, se podría hacer un reajuste de estos para adaptarla mejor a las necesidades. Además el diseño de la placa se podría simplificar haciendo que fuera más compacto y menos tosco.
- Los protocolos de comunicación entre la interfaz Nao y el Line printer están implementados sobre el propio código de éstos, lo cual no es muy buena implementación. Habría

que reestructurar de alguna manera este software para que tuviera mayor claridad y orden y fuera un único componente software compartido entre Nao y el Line printer el que implementara el protocolo de comunicación entre estos.

- Si bien es cierto que la Raspberry es muy poderosa, al ser una maquina de ámbito general esta tiene ligada su ejecución al planificador de tareas, el cual puede expropiar en cualquier momento a cualquier hilo de ejecución en el pipe. Esto representa un grave problema durante la ejecución de una impresión que podría generar imprecisiones en ésta. Este posible problema se puede arreglar con el uso de un hardware dedicado como puede ser el de Arduino.

Aun así, también tiene sus partes positivas:

- El PLA es un material biodegradable lo cual permite no solo su fácil eliminación de este cuando se considere residuo sino la reutilización para otras impresiones. Por el momento es la opción más ecológica para imprimir.
- El protocolo de comunicación es modular, escalable y portable. Es decir, permite que la aplicación Nao se integre con cualquier otro software para controlar impresoras del mismo modo que el Line printer, requeriría de pocas modificaciones para llevarlo a cabo. También permite controlar varias impresoras simultaneamente e ir añadiendo más impresoras a la aplicación y este software esta adaptado para que se pueda ejecutar en cualquier red. Eso quiere decir que si trasladamos el PC del usuario y la impresora a otra red, la configuración seguirá siendo válida y no tendremos que volver a añadir la impresora, podríamos controlarla como si no hubiéramos cambiado de red.
- Slicer permite desligarse totalmente del G-code el cual es un estándar de código en el que cada implementación esta ligada a un hardware específico, esto es un impedimento para poder controlar impresoras cualesquiera con softwares cualesquiera. Slicer trabaja solo con cosas que todas las impresoras tienen en común, coordenadas medidas en pasos de motor, eso quiere decir que este, con pocas adaptaciones, podría controlar cualquier impresora.
- Con la implementación del Line printer y la de Slicer se palia el problema de la expropiación del pipe haciendo que aunque esto suceda como las distancias entre puntos son tan

pequeñas el error sea imperceptible para el ojo humano. Además un sistema de ámbito general como es la Raspberry permite que en el futuro se puedan añadir más tareas independientes a la impresión. De otro modo tendríamos que ir añadiendo más hardware de control adicional para cada nueva funcionalidad que queramos darle a la impresora.

- La estructura imprime figuras de medidas cercanas a su tamaño total, es decir, que teniendo unas dimensiones de 50x50x50 centímetros, aprovecha el espacio de este bastante más que lo que hacen otras muchas impresoras las cuales tienen una relación tamaño de impresión/tamaño bastante deficitario.
- Como las figuras impresas no se apoyan sobre ninguno de los ejes esto permite que sea escalable, es decir, que como el tamaño de la figura es independiente, esta estructura se podría replicar a los tamaños que deseemos.

Trabajos futuros:

- Adaptar la impresora para añadir nuevos materiales de impresión
- Añadir la funcionalidad de un relleno automático de figuras en el Slicer.
- Añadir la funcionalidad de construir estructuras de soporte automáticamente durante la impresión en el Slicer.
- Implementar mejor la funcionalidad de poder controlar más de una impresora en Nao.
- Añadir hardware, controlado por la raspberry, para que esta se encargará de la impresión y dejar más libertad a la raspberry para realizar otras tareas. Este hardware podría ser una Arduino.
- Crear un software de modelado de figuras 3D que se adapte mejor a la implementación actual de la impresora.
- Adaptar mejor el Slicer para que pueda trabajar con más formatos de figuras tridimensionales que el .stl.
- Crear un sistema para poder controlar la impresora sin la necesidad de usar la interfaz Nao, directamente sobre el PC de la impresora.

Conclusions

3D printing is a field in full boil which does not stop advancing day by day, new advances, printing techniques and findings make this technology is increasingly powerful and has greater versatility making it useful in virtually all fields of the society. It seems that it is the road to which the future points and this seems promising.

During the development of the project, as it progressed, I became aware of the difficulty that unravels the performance of this technology, there are many decisive factors that determine whether or not the implementation of a printer works and the probability of problems arising is high. This combined with the little previous knowledge in the matter is what has slowed down the performance of this work with greater degree.

This project had the following objectives:

- Develop my engineering skills acquired during the race and test these.
- Lay the foundations of a software-hardware design to extend this technology.
- Enhance this technology.

Of these, the first was fulfilled many beyond my expectations, the knowledge acquired during the development of this project has been of very large proportions, both in matters related to the career and in other areas outside of this such as:

- Significantly improve my ability to reason and deduction in solving problems.
- Help to structure my head in a more orderly and efficient way, resulting in an orderly and logical project that follows some firm steps in its development.
- Improve my ability to estimate how long each task would take and how complex it was. Which resulted in an improvement in my ability to manage project times.

As for the other two objectives, we can say that they were met but with certain reluctance. It is true that I have managed to develop the printer but for the moment it is far from being a fully viable option to the current alternatives. There is still a lot of work ahead so that this is an option as valid as the alternatives. These are the drawbacks or weakest parts of the project:

- Only allows you to print on PLA, you do not have a system ready to print on other more solid and resistant materials.
- The x axis has certain mechanical problems, sometimes the linear guides of the axis skid and this produces precision errors in the printing.
- The z-axis dances due to the little friction between the zipper guide and the rack itself, causing that when there are small vibrations move the axis and produce errors of precision in the printing.
- The Slicer software is not as efficient as it could be, it has several better possible to reduce the computational cost in its execution. In addition, it has no option to automatically fill the figures according to the needs of the user or automatically generate support structures to create cantilevered figures as other software such as the Cure or the Slic3r, which makes it much less powerful than their counterparts.
- The board has elements over dimensioned for the required needs which represents an over cost, you could make a readjustment of these to adapt it better to the needs. In addition the design of the plate could be simplified making it more compact and less rough.
- The communication protocols between the Nao interface and the Line printer are implemented on their own code, which is not a very good implementation. Somehow this software would have to be restructured so that it would have greater clarity and order and it would be a single shared software component between Nao and the Line printer that would implement the communication protocol between them.
- While it is true that the Raspberry is very powerful, as it is a general-purpose machine, it has its execution linked to the task scheduler, which can expropriate at any time any execution thread in the pipe. This represents a serious problem during the execution of an

impression that could generate inaccuracies in it. This possible problem can be fixed with the use of a dedicated hardware such as Arduino.

Even so, it also has its positive parts:

- The PLA is a biodegradable material which allows not only its easy elimination when it is considered waste but reuse for other impressions. At the moment it is the greenest option to print.
- The communication protocol is modular, scalable and portable. That is, it allows the Nao application to be integrated with any other software to control printers in the same way as the Line printer, it would require few modifications to carry it out. It also allows to control several printers simultaneously and to add more printers to the application and this software is adapted so that it can be executed in any network. That means that if we move the user's PC and the printer to another network, the configuration will remain valid and we will not have to add the printer again, we could control it as if we had not changed the network.
- Slicer allows to completely separate from the G-code which is a code standard in which each implementation is linked to a specific hardware, this is an impediment to be able to control any printers with any software. Slicer works only with things that all printers have in common, coordinates measured in steps of motor, that means that this, with few adaptations, could control any printer.
- With the implementation of the Line printer and the one of Slicer, the problem of the expropriation of the pipe is alleviated, even though this happens as the distances between points are so small the error is imperceptible to the human eye. In addition, a general-purpose system such as Raspberry allows more independent tasks to be added to the printing in the future. Otherwise we would have to add more additional control hardware for each new functionality that we want to give to the printer.
- The structure prints figures of measures close to their total size, that is, having a size of 50x50x50 centimeters, it takes advantage of the space of this much more than what many other printers do, which have a ratio size of impression / size deficit.

- As the printed figures do not rest on any of the axes this allows it to be scalable, that is, since the size of the figure is independent, this structure could be replicated to the sizes that we want.

Future works:

- Adapt the printer to add new printing materials
- Add the functionality of an automatic filling of figures in the Slicer.
- Add functionality to build support structures automatically during printing on the Slicer.
- Better implement the functionality of being able to control more than one printer in Nao.
- Add hardware, controlled by the raspberry, so that this will take care of the printing and leave more freedom to the raspberry to perform other tasks. This hardware could be an Arguino.
- Create a 3D figure modeling software that best suits the current printer implementation.
- Better adapt the Slicer so you can work with more three-dimensional shape formats than the .stl.
- Create a system to control the printer without the need to use the Nao interface, directly on the printer PC.

Glosario

3DP 3D Printing.

aglutinante Se dice de un material que es capaz de unir distintos elementos entre si.

BPM Ballistic Particle Manufacturing.

DIY Do it yourself.

estereolitografía Proceso de fabricación por adición que emplea resina que cura mediante luz ultravioleta en un tanque, y un láser ultravioleta para construir los objetos. Los objetos tridimensionales son obtenidos mediante la adición de finas capas, impresas una encima de otra. Cada capa es una sección transversal del objeto que el láser traza en la superficie de la resina, que es el material consumible. La resina líquida cura y se solidifica mediante la exposición al láser de luz ultravioleta, quedando así la capa recién solidificada pegada a la capa previa que existía debajo suya.

extruir Proceso por el que se moldea un material y se aplana, se comprime al punto de darle forma de lámina o cualquier otra forma. En el caso de las impresoras 3D se moldea este material aplicándole calor.

FDM Fused Deposition Modelling.

fotopolímero Un fotopolímero es una sustancia sintética que sufre un cambio en sus propiedades (polimerización, reticulación o despolimerización) por acción de la luz, generalmente ultravioleta, formando una diferenciación física entre las partes expuestas y no expuestas.

LOM Laminated Object Manufacturing.

MakerFaire Es un evento en el que se reúnen los creadores de inventos y proyectos para mostrárselos entre sí. La impresión 3D suele ser uno de los temas de las “maker faires” junto con la robótica y otras tecnologías emergentes.

modelo 3D Es el producto resultante del proceso de modelación 3D. Este proceso se define como el proceso de desarrollo de una representación matemática de cualquier objeto tridimensional (ya sea inanimado o vivo) a través de un software especializado.

propiedad mecánica Son propiedades físicas que describen el comportamiento de un material sólido al aplicarle fuerzas de tracción, compresión y torsión.

RepRap Proyecto creado con la finalidad de crear maquinas autorreplicables que se puedan usar para prototipado rápido y manufactura. Este es un movimiento que tiene por objetivo potenciar la tecnología 3D.

sinterización Proceso de fabricación de piezas sólidas moldeadas, consiste en compactar a alta presión varios polvos metálicos y/o cerámicas mezcladas homogéneamente y, una vez compactadas, realizar un tratamiento térmico, a una temperatura inferior a la de fusión de la mezcla, obteniéndose una pieza consolidada y compacta.

SLS Selective Laser Sinteting.

voladizo Elemento que se prolonga horizontal o inclinadamente más allá de la base que lo soporta.

Para más información [48].

Bibliografía

- [1] Precio medio de las impresoras.
- [2] Sketchup.
- [3] EasyEda.
- [4] RaspberryPi.
- [5] Eclipse.
- [6] OpenScad.
- [7] Ubuntu.
- [8] Reporte de ventas de impresoras 3D del año 2016.
- [9] Explicación de la técnica SGC.
- [10] Explicación de la técnica EBM.
- [11] Tinta sólida.
- [12] Polimerización térmica.
- [13] Tipos de filamentos más usados en impresión 3D.
- [14] Explicación del funcionamiento de un motor de continua.
- [15] SMRC.
- [16] Natural machines.
- [17] 3DSystems.

- [18] MIT.
- [19] Nike.
- [20] Airbus.
- [21] Agencia espacial Europea.
- [22] NASA.
- [23] BOEING.
- [24] GE aviation.
- [25] Escudería Williams.
- [26] EOS.
- [27] Local motors.
- [28] Nervous system.
- [29] Strata systems.
- [30] Materialise.
- [31] Universidad de Michigan.
- [32] Universidad Wake Forest.
- [33] Organovo.
- [34] Instituto tecnológico de Canarias.
- [35] Podoactiva.
- [36] Magic Arms.
- [37] Enabeling the future.
- [38] Hoja de datos de la RaspberryPi.
- [39] Explicación del funcionamiento de un motor bipolar.

- [40] Hoja de datos del motor Nema 17 modelo 42BYGHM809.
- [41] Hoja de datos del motor Nema 17 modelo 42hd4027-01.
- [42] Hoja de datos del variador DRV8825.
- [43] Hoja de datos del convertidor analógico-digital MCP3002.
- [44] Hoja de datos del relé de estado sólido CMX100D10.
- [45] Explicación del funcionamiento de un relé de estado sólido.
- [46] Hoja de datos del termistor 100K NTC.
- [47] MPFR.
- [48] Diccionario de términos relacionados con la impresión 3D.
- [49] Dickerson, Matthew T. En «*Fast greedy triangulation algorithms*». Elsevier, 1997. <https://www.sciencedirect.com/science/article/pii/S0925772197891493>.
- [50] González, Sergio Gómez. En «*Impresión 3D*». Marcombo, Junio 2016. <https://www.marcombo.com/impresion-3d-9788426723536/>.
- [51] Loera, Jesús A., Joerg Rambau y Franciso Santos. Springer Science and Business Media.
- [52] Portero, Juan Carlos Arroyo, Ramón Sánchez Fernández, Antonio Romero Ballesteros, Manuel G. Romana, Guillermo Corres Peiretti y Gonzalo García-Rosales. En «*Números gordos en el proyecto de estructuras*». Cinter, 2001. <https://www.cinter.es>.